

(19) 日本国特許庁 (J P)

(12) 公表特許公報 (A)

(11) 特許出願公表番号

特表2002-501251

(P2002-501251A)

(43) 公表日 平成14年1月15日 (2002.1.15)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テマコード* (参考)
G 0 6 F 13/00	3 5 7 3 5 1	G 0 6 F 13/00	3 5 7 A 5 B 0 8 9 3 5 1 B 5 K 0 3 3
H 0 4 B 3/54		H 0 4 B 3/54	5 K 0 4 6
H 0 4 L 12/46		H 0 4 L 11/00	3 1 0 C
12/28			

審査請求 未請求 予備審査請求 有 (全128頁)

(21) 出願番号 特願2000-528920(P2000-528920)  
(86) (22) 出願日 平成11年1月21日 (1999.1.21)  
(85) 翻訳文提出日 平成12年7月24日 (2000.7.24)  
(86) 国際出願番号 PCT/US 99/01234  
(87) 国際公開番号 WO 99/38084  
(87) 国際公開日 平成11年7月29日 (1999.7.29)  
(31) 優先権主張番号 60/072,128  
(32) 優先日 平成10年1月22日 (1998.1.22)  
(33) 優先権主張国 米国 (US)

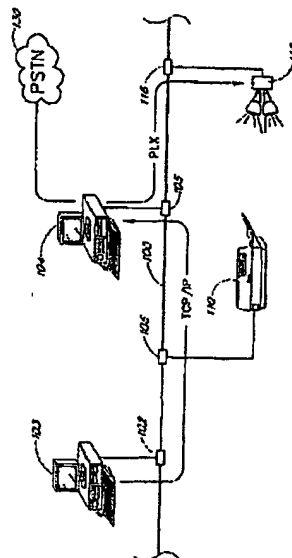
(71) 出願人 イナリ、インコーポレイテッド  
INARI, INC.  
アメリカ合衆国 84020 ユタ州 ドレー  
パー スイート 100 サウス ローン  
ピーク パークウェイ 11781  
(72) 発明者 ワルベック、 アラン  
アメリカ合衆国 84093 ユタ州 サンデ  
イー イー、 プラータ ウェイ 1746  
(72) 発明者 リー、 トーマス、 エヌ.  
アメリカ合衆国 84660 ユタ州 スパニ  
ッシュ フォーク イー、 850 サウス  
1051  
(74) 代理人 弁理士 金田 暢之 (外2名)

最終頁に続く

(54) 【発明の名称】 ユニバーサル・データ交換ゲートウェイのための方法および装置

(57) 【要約】

1つまたは複数のネットワーク・プロトコルと、1つまたは複数の制御プロトコルの間でデータを転送できるようにするユニバーサル・ゲートウェイ (104) が記載されている。種々のプロトコルが同じ物理的ネットワーク媒体 (100) または別々のネットワーク (100、130) に存在できる。ゲートウェイ (104) を用いることによって、エンドユーザーは従来の自立した、互換性のないネットワークを一緒に結合して、ユニバーサルにアクセスできる、中央管理された「スーパーネットワーク」にできる。ゲートウェイ (104) は集中化されたノード・データベースと、レガシー・プロトコルのサポートと、規則エンジンと、オブジェクト指向クラス・ライブラリ・インタフェースとを提供する。集中化されたノード・データベースに対する高信頼性のアクセスが、待機サーバ・ノードによって提供されるシステム障害耐性により強化される。また、ゲートウェイ (104) は各種のデータストリームを電源線を通じて分配する能力をもたらし。ゲートウェイ (104) によって提供された経路選択ハンドラによりTCP/IPなどの、



**【特許請求の範囲】**

【請求項1】 媒体プロトコルを用いてネットワーク媒体上を伝送させられる1つまたは複数のデータ・プロトコルを用いてコンピュータ・ネットワーク上の多数のノードが通信できるように構成され、前記多数のノードと通信するためにアプリケーション・プログラミング・インタフェースをさらに提供するゲートウェイであって、

ネットワーク上ノードについての情報を含む内部ノード・データベースと、  
前記内部ノード・データベースを維持して、前記ノード・データベースをアクセスするように構成されているアクティブモードと、前記内部ノード・データベースを外部ノード・データベースの鏡像コピーとして維持するように構成されている待機モードとを提供するように構成されているソフトウェア・モジュールとを有するゲートウェイ。

【請求項2】 前記内部ノード・データベースが、クライアント・ノードの状態変化時にとるべき行動を指定する規則をさらに含む、請求項1に記載のゲートウェイ。

【請求項3】 前記規則が簡単な規則である、請求項2に記載のゲートウェイ。

【請求項4】 前記規則が複雑な規則である、請求項2に記載のゲートウェイ。

【請求項5】 前記規則を解釈するように構成された規則エンジンをさらに備える、請求項2に記載のゲートウェイ。

【請求項6】 シムをさらに備え、前記シムは規則を規則定義言語に変換するように構成されている、請求項2に記載のゲートウェイ。

【請求項7】 前記状態変化が前記クライアント・ノードのインスタンス変数の変化を含む、請求項2に記載のゲートウェイ。

【請求項8】 ピン要求を発することによって前記内部ノード・データベースが更新される、請求項1に記載のゲートウェイ。

【請求項9】 前記ソフトウェア・モジュールが、確認されていないクライアント要求が検出された時に、前記アクティブモードへ移行するように構成され

ている、請求項1に記載のゲートウェイ。

【請求項10】 第1のプロトコルを第2のプロトコル内を通りぬけさせるようにさらに構成されている、請求項1に記載のゲートウェイ。

【請求項11】 前記媒体が電源線であり、前記媒体プロトコルが電源線プロトコルである、請求項10に記載のゲートウェイ。

【請求項12】 前記媒体が電源線であり、前記媒体プロトコルがPLXプロトコルである、請求項1に記載のゲートウェイ。

【請求項13】 前記クライアント・ノードのインスタンス変数に変化が起きた時にユーザー・アプリケーションに通知するように構成されている事象ハンドラをさらに備える、請求項7に記載のゲートウェイ。

【請求項14】 オブジェクト指向アプリケーション・プログラミング・インタフェースをさらに備える、請求項1に記載のゲートウェイ。

【請求項15】 ユーザー・インタフェースに前記内部ノード・データベース中の情報を提供するように構成されているインターネット・ブラウザをさらに備える、請求項14に記載のゲートウェイ。

【請求項16】 ユーザーが電源線ネットワーク上のノードを制御できるようにするように前記ユーザー・インタフェースが構成されている、請求項15に記載のゲートウェイ。

【請求項17】 電源線ネットワーク媒体と、  
ディスパッチ制御ブロックを使用することによって生のデータ情報を、前記電源線ネットワーク媒体からユーザー・アプリケーションへ送るゲートウェイ手段とを有するコンピュータ・ネットワーク。

【請求項18】 ノード・データベースと、  
ディスパッチ制御ブロックを作成して解釈するディスパッチャ手段と、  
ネットワーク・インタフェース・アダプタを制御するデバイス・ドライバ手段と、

前記デバイス制御ブロックを前記デバイス・ドライバ手段に適合させるシム手段とを有するゲートウェイ。

【請求項19】 ネットワーク上のノードについての情報を含んでいるノー

ド・データベースを作成することと、

前記ノード・データベースをアクセスする1つまたは複数のアクセス法を提供し、前記ノード・データベースを維持するゲートウェイを指定することと、

前記ノード・データベースを1つまたは複数の待機サーバ・ノード内にすることと、

を有する、ネットワーク上のノード間で通信するために所望のプロトコルを使用する方法。

【請求項20】 クライアント・ノードにおいて状態変化が起きた時取るべき行動を指定する諸規則を解釈し、かつ実行することをさらに備える請求項19に記載の方法。

【請求項21】 前記諸規則が規則エンジンによって解釈される、請求項20に記載の方法。

【請求項22】 前記状態変化が起きた時に事象通知を発生するステップをさらに有する、請求項20に記載の方法。

【請求項23】 前記通知がディスパッチャに与えられる、請求項22に記載の方法。

【請求項24】 受けとったデータを規則定義言語に変換するステップをさらに有する、請求項20に記載の方法。

【請求項25】 前記状態変化が前記クライアント・ノードのインスタンス変数の変化を含む、請求項20に記載の方法。

【請求項26】 ピン要求を発し、前記ピン要求に対する応答が聞こえるかと耳を澄ますステップをさらに有し、前記応答は前記ノード・データベースを更新するために用いられる、請求項19に記載の方法。

【請求項27】 前記アクティブサーバがインアクティブになった後に前記待機サーバの1つをアクティブ状態にするステップをさらに有する、請求項19に記載の方法。

【請求項28】 第1のプロトコルの生のパケットを前記所望のプロトコルでの包みパケット内に入れ、前記生のパケットを前記所望のプロトコル内に通りぬけさせるステップとをさらに有する、請求項19に記載の方法。

【請求項29】 前記媒体が電源線であり、前記媒体プロトコルが電源線プロトコルである、請求項19に記載の方法。

【請求項30】 前記媒体が電源線であり、前記媒体プロトコルがPLXプロトコルである、請求項19に記載の方法。

【請求項31】 前記クライアント・ノードのインスタンス変数が変化した時にユーザー・アプリケーションに通知するステップをさらに有する、請求項19に記載の方法。

【請求項32】 前記ノード・データベース内の情報を見るためにインターネット・ブラウザを用いるステップをさらに有する、請求項19に記載の方法。

【請求項33】 電源線ネットワーク上のノードを制御するためにインターネット・ブラウザを用いるステップを更に備える請求項19に記載の方法。

## 【発明の詳細な説明】

## 【0001】

## 【技術分野】

発明の背景発明の分野

本発明はコンピュータ・ネットワーク・プロトコル・ゲートウェイに関し、特に、電源線ネットワーキング・システムに適合させられるゲートウェイに関する。

## 【0002】

## 【従来の技術】

関連技術についての説明

コンピュータ、特にパーソナルコンピュータの広範囲な可用性によってコンピュータ・ネットワークの数が急速に増加した。2台または3台以上のコンピュータを一緒にネットワーク化することによってコンピュータが情報、ファイル資源、プリンタ等を共用できる。2台または3台以上のパーソナル・コンピュータおよびプリンタを一緒に接続してネットワークを構成することは簡単な作業である。コンピュータとプリンタはケーブルを用いて一緒に簡単に接続され、必要なソフトウェアがコンピュータにインストールされる。ネットワークの用語では、ケーブルはネットワーク媒体であり、コンピュータとプリンタはネットワーク・ノードである。ネットワーク・ノードは、伝送制御プロトコル、インターネット・プロトコル(TCP/IP)などのプロトコルを1つまたは複数個用いて相互に「話しあう」。

## 【0003】

ゲートウェイは1つのプロトコルから別のプロトコルへのプロトコル翻訳を実行するために用いられるので、種々のプロトコルを使用する2つのネットワークを相互に接続できる。たとえば、Prodigyネットワーク・サービスは、その内部に所有する電子メール・フォーマットとインターネット電子メールフォーマットとの間で翻訳するゲートウェイを有する。

## 【0004】

標準的なネットワーク・プロトコルは通常、各ネットワーク・ノードが、十分な処理性能と十分な記憶性能を持つ「スマート (s m a r t)」な装置であるとの仮定の下に作られている。たとえば、典型的なパーソナル・コンピュータ (P C) はほとんどいかなるネットワーク・プロトコルも取り扱う十分以上の処理性能と記憶性能を有する。しかし、プリンタは必要な処理性能と記憶性能を有しない「ダム (d u m b)」装置である。プリンタをネットワークに接続できるようにするネットワーク・プリンタ・アダプタを提供する製造業者もいる。プリンタ・アダプタは、完全に構成された P C の処理性能と記憶性能に類似する処理性能と記憶性能を提供する単一のボードコンピュータである。したがって、ネットワーク・プリンタ・アダプタは「ダム」プリンタを「スマート」装置に変換する。ネットワーク・プリンタ・アダプタは実際に動作するが、比較的高価であるので多くの家庭や小規模の事務所の環境には向いていない。さらに、プリンタ・アダプタはその他の非 P C 装置をネットワークに接続するのにはあまり良く適していない。たとえば、ユーザーは、屋外灯、警報器、電話装置等などのダム装置をユーザーのコンピュータ・ネットワークに接続することをしばしば希望する。それらの各ダム装置をスマート装置の変えるネットワーク・アダプタ・カードを購入することは、それを断念させるほど高価につく。

#### 【0005】

スマート装置のために使用されるプロトコルは「ネットワーク・プロトコル」と通常呼ばれている。ダム装置のために用いられるプロトコルは「制御プロトコル」としばしば呼ばれる。ネットワーク・プロトコルは、性能と複雑さとにおいて、制御プロトコルとは全く異なる。それらの違いのために、ネットワーク・プロトコルの間でデータを転送するように構成されているゲートウェイは、制御プロトコルの間でデータを転送するという作業には通常あまり適さない。より困難なことはネットワーク・プロトコルと制御プロトコルの間でデータを転送する作業である。

#### 【0006】

既存の家庭用制御／オートメーション製品は、集中化されたクライアント／サーバ・モデルではなくて同列間モデルを基にしている制御プロトコルを使用する

傾向がある。これは、各ノードが状態情報と諸規則をローカルに記憶することが要求されるために、これらの製品の可用性を制限する。使用が容易な、集中化されたユーザー・インタフェース部品がないために、ネットワークの構成はしばしば困難である。さらに、競合する製品（たとえば、X-10とCEBus）の間の相互使用可能性がほとんど不可能である。

#### 発明の概要

本発明は、1つまたは複数のネットワーク・プロトコルと1つまたは複数の制御プロトコルの間でのデータ転送可能にする、安価で、使用が容易であり、融通性があり、信頼性が高く、かつ拡張性があるゲートウェイ・アーキテクチャを提供することによって、それらの問題およびその他の問題を解決する。種々のプロトコルが同じ物理的ネットワーク媒体上に共存できる。このゲートウェイは、選択されたプロトコルの中にネットワーク・プロトコルを通り抜けさせ（`tunneling`）、かつ集中化された制御も行う。

#### 【0007】

したがって、このゲートウェイはデータおよび制御ネットワークのエンドユーザー、特に家庭および小規模事務所の環境における利用者に極めて大きな利益をもたらす。このゲートウェイによって、エンドユーザーは、他のネットワークと非互換の従来の自立したネットワークを容易かつ好都合に組み合わせて、コンピュータ、プリンタ、警報装置、家庭用機器、照明装置、電話等を接続する普遍的にアクセスできる、集中管理される「スーパーネットワーク」にできる。

#### 【0008】

このゲートウェイは集中化されたノード・データベースと、TCP/IPなどのレガシー（`legacy`）プロトコルの支持、規則エンジン、およびオブジェクト指向のクラス・ライブラリ・インタフェースを提供する。このゲートウェイは、インターネット・ブラウザなどの使用が容易で、共通のグラフィカル・ユーザー・インタフェースを用いて家庭用／小規模事務所のオートメーションおよび制御の広域な可能性を提供する。自動装置の発見によって構成が簡単にされる。集中化されたノード・データベースに対するアクセスは、待機サーバによって提供されるシステム障害耐性によって高められる。



## 【0009】

このゲートウェイは、電源線ネットワークと共に使用されると、各種のデータストリームを電源線を通じて分配する能力を有する。たとえば、ケーブルモデムまたはその他の種類の高速インターネット接続器を有するネットワーク・ユーザーは、既に設置されている電気配線以外の追加の配線の必要なしに家庭／事務所内のどこにでもインターネット・トラフィックを送ることができる。音声データも電源線を通じて家庭／事務所全体に容易に送られる。このゲートウェイにより提供されるハンドラによって、今日一般に用いられているほとんど全てのレガシイ・データネットワーク化サービスおよびプロトコルを、電源線を通じて送ることができる。

## 【0010】

開示されている発明の諸利点および諸特徴は、当業者は下記の図面とともに以下の詳細な説明を読んだ時に容易に理解されるであろう。

## 【0011】

図面において、任意の3桁の番号の最初の数字は、要素が最初に現れた図の番号を一般に示している。4桁の参照番号が用いられた時は、初めの2桁が図面番号を示す。

好適な実施形態の詳細な説明

図1はユニバーサル・ゲートウェイとともに使用されるのに適したコンピュータ・ネットワーク・システムを示す。第1の物理的ネットワークはネットワーク媒体100（ケーブルとして示されている）を用いている。スマート・ノード（パーソナル・コンピュータ103として示されている）はコネクタ102によってネットワーク媒体100に接続されている。プリンタ110と、コンピュータ104と、安全照明装置118もネットワーク媒体100に接続されている。照明装置118は、比較的低い計算能力と比較的小さい記憶を持つ「ダム」ノードの例である。コンピュータ104は、公衆交換電話回線（PSTN）として示されている第2のネットワーク130にも接続されている。

## 【0012】

典型的なネットワーク環境においては、ネットワーク媒体100は各種のデー

タ・プロトコルのためのデータ・トラフィックを伝えるように構成されている。したがって、図1における例により示されているように、コンピュータ103はTCP/IPを用いてコンピュータ104と通信でき、コンピュータ104は、たとえば、この開示された出願の、付録Aとして表示されている部分に記載されている電源線交換(PLX)プロトコルなどの制御プロトコルを用いて照明装置118と通信する。付録Aは「電源線交換プロトコル方法および装置(METHOD AND APPARATUS FOR POWER LINE EXCHANGE PROTOCOL)」という名称の米国特許出願09/211950の部分を含む。この米国特許出願の開示は参照することによりここに含まれる。

#### 【0013】

ユニバーサル・ゲートウェイは、コンピュータ104などのコンピュータでソフトウェア・プログラムとして実行される。このユニバーサル・ゲートウェイは種々のネットワーク・プロトコルの間、および種々の物理的ネットワークの間の接続を行う。たとえば、図1に示すように、コンピュータ104においてソフトウェア・プログラムとして実行されているこのユニバーサル・ゲートウェイはTCP/IPプロトコルをPLXプロトコルに接続することにより、コンピュータ103が照明装置118と通信できるようにする。

#### 【0014】

コンピュータ104においてソフトウェア・プログラムとして実行されているこのユニバーサル・ゲートウェイは別々の物理的ネットワークの間のデータ転送も行うことにより、別々の物理的ネットワークにおける装置が相互に通信できるようにする。図1において、このユニバーサル・ゲートウェイはネットワーク130と照明装置118との間のデータ転送を行う。

#### 【0015】

このユニバーサル・ゲートウェイは階層状のネットワーク・プロトコルに適合する。スマート・ノード(コンピュータ103および104など)用に構成されたほとんどのネットワークは、オープン・システム・インタフェース(OSI)委員会により開発されたネットワーク・アーキテクチャ・モデルを基にしている。OSIアーキテクチャは、通信システム内の個々の各ハードウェア層およびソ

フトウェア層と、層の間の相互依存性と、各層が実行する独自の機能とについての概要を示すネットワーク・モデルを定める。

#### 【0016】

図2は、1番下の層から1番上の層まで、物理層201、データ・リンク層202、ネットワーク層203、トランスポート層204、セッション層205、プレゼンテーション層206、およびアプリケーション層207の7つの層としてOSIアーキテクチャが編成されていることを示す。各層はそのすぐ下の層を用いてすぐ上の層のサービスを行う。ある実現例では、ある層は副層により自身で構成できる。ある層は、特定のネットワーク・プロトコルが動作する2つまたは3つ以上の通信装置またはコンピュータの、ソフトウェア環境および／またはハードウェア環境である。ネットワーク接続は、おのおの異なる層またはレベルにある、多少とも独立している1組のプロトコルであるとして考えることができる。1番下の層は、種々のノードにおけるハードウェアの間の直接ノード間通信を支配し、1番上はユーザー・アプリケーション・プログラムで構成されている。各層はその下の層を用いて上の層に対するサービスを行う。1つのホストにおける各ネットワーク化コンポーネント・ハードウェアまたはソフトウェアは、その層に適切なプロトコルを用いて、他のノードにおける対応するコンポーネント（その「同等なコンポーネント」）と通信する。そのような階層状のプロトコルは同列間プロトコルとして知られていることがある。

#### 【0017】

階層状にされたプロトコルの利点は、1つの層から別の層へ情報を送る方法がプロトコルの部分として明らかに指定され、かつ1つのプロトコル層内での変化が他のプロトコル層に影響を及ぼすことが阻止されることにある。これによって通信システム的设计および維持が簡単になる。

#### 【0018】

物理層201はOSI階層化モデル内の1番下の層である。それは媒体アクセス制御(MAC)を含むネットワークの電氣的接続と機械的接続を含む。媒体アクセス制御とは、データ伝送媒体100（たとえば、ネットワーク・ケーブル）の制御、およびそれに対するアクセスを指す。物理層201はデータ・リンク層

202により用いられる。

#### 【0019】

データ・リンク層202はOSI層モデル内の1番下から2番目の層である。データ・リンク層202はデータを（物理層201で送られる）フレームに分割し、確認応答フレームを受信する。データ・リンク層202は誤り検査を行い、正しく受信されなかったフレームを再送する。データ・リンク層202は誤りのない仮想チャネルをネットワーク層203に提供する。データ・リンク層202は上側の副層、すなわち、論理リンク制御（LLC）と、媒体アクセス制御（MAC）の部分を含む下側の副層とに通常分割される。

#### 【0020】

ネットワーク層203はOSIの7層モデル内の1番下から3番目の層である。ネットワーク層203は、送り手からデータ・リンク層202を経由する受け手までのデータ・パケットの経路を決定し、かつトランスポート層204によって用いられる。最も一般的なネットワーク層プロトコルはIPである。

#### 【0021】

トランスポート層（または「ホスト-ホスト層」）204はOSIモデル内の中間層である。トランスポート層204は、最初のノードがメッセージを第2のノードへ送ることができ、それらのメッセージが変更されないで、正しい順序で到達するように、誤りのない点間の仮想接続を行うためにネットワーク層203をどのように使用するかを決定する。トランスポート層204はノードの間の接続を行い、かつ解消する。

#### 【0022】

セッション層205はOSIモデル内の1番上から3番目のプロトコル層である。セッション層205はトランスポート層204を用いて種々のノードにおけるプロセス間の接続を確立する。セッション層205はセッションの安全と作成を取り扱う。

#### 【0023】

プレゼンテーション層206はOSIモデル内の1番上から2番目のプロトコル層である。プレゼンテーション層206はノードの間の違いをならそうとして

テキスト圧縮、コードまたはフォーマットの変換などの機能を実行する。プレゼンテーション層206によってアプリケーション層内の適合しないプロセスがセッション層を介して通信できるようにされる。

#### 【0024】

アプリケーション層207はOSIモデル内の1番上の層である。アプリケーション層207はネットワークについてユーザーが見るもの（たとえば、電子メール・メッセージのフォーマット）に関するものである。プレゼンテーション層206はアプリケーション層207に、ネットワークで使用されているフォーマットとは独立したデータのなじみのローカル表現を提供する。アプリケーション層プロトコル例にはTelnet、ファイル転送プロトコル（FTP）、シンプル・ネットワーク・マネージメント・プロトコル（SNMP）、シンプル・メール転送プロトコル（SMTP）、インターネット制御メッセージ・プロトコル（ICMP）、ネットウェア・コア・プロトコル（NCP）、経路指定情報プロトコル（RIP）、サービス・アドバータイジング・プロトコル（SAP）、単純ファイル転送プロトコル（TFTP）、およびシステムフォールトトレランスプロトコル（SFTP）が含まれる。

#### 【0025】

図3はユニバーサル・ゲートウェイ300のアーキテクチャと、ゲートウェイ300が、アプリケーション・プログラム、ハードウェア・ドライバなどの他のソフトウェア・コンポーネントとどのようにしてやり取りするかを示す。アプリケーション・プログラム302は1組のゲートウェイ・ファウンデーション・クラス304を用いてゲートウェイ300と通信する。ゲートウェイ・ファウンデーション・クラス304はデータベース・アクセス・モジュール306と、事象ハンドラ310と、ディスパッチャ320と、ルーティンオペレーティングシステム・サービス（RTOS）311と通信する。データベース・アクセス・モジュール306は、（ノード・データベース）と呼ばれている情報保存部308を含む。ノード・データベース308はデータベース、リンクされたリスト、表、マップ、コンテナ等として編成できる。データベース・アクセス・モジュール306は事象ハンドラ310と、ペイロード／プロトコル・プロセッサ312とも

通信する。ペイロード／プロトコル・プロセッサ312は、生データ・ハンドラ316、ストリーミング・データ・ハンドラ317、およびCAL制御ハンドラ318などのペイロード／プロトコル・ハンドラを1つまたは複数個含む。事象ハンドラ310は規則エンジン314およびディスパッチャ320とも通信する。ディスパッチャ320はRTOS311と、規則エンジン314と、PLX装置ドライバ322と、レガシイデバイスドライバ326とも通信する。PLXデバイスドライバ322はPLX制御装置ドライバ323（ダム装置のため）とPLXデータ装置ドライバ324（スマート装置のため）を含む。

#### 【0026】

ゲートウェイ300は1つのプロトコルから別のプロトコルへのプロトコル翻訳を行うので、異なるプロトコルを使用している2つのネットワークを相互に接続できる。それらのネットワークは実在の物理的ネットワークとすることもできれば、ネットワークを仮想ネットワーク（すなわち、ソフトウェア内にのみ存在する）とすることができる。ゲートウェイ300はレガシイ・プロトコルと主（すなわち、所望の）プロトコル（たとえば、電源線プロトコル）との間のインタフェースを行う。レガシイ・プロトコルという用語は既存のプロトコルに限定されるものではない。それよりも、レガシイ・プロトコルという用語は主プロトコル以外の任意のプロトコルを指すことを意図するものである。レガシイデバイスドライバは、たとえば、X-10ドライバ329とCEBusドライバ332を含む。各レガシイ装置ドライバは、レガシイ装置ドライバをゲートウェイに適合させるシムを含むことを選択もできる。一実施形態では、主プロトコルは電源線プロトコルであるが、ゲートウェイ300はそのようには限定されない。したがって、主プロトコルは、たとえば、TCP/IP、IPX、ADSL、およびこの開示の他の箇所に掲記されているか、この技術で知られているその他のプロトコルのいずれかを含めた任意のプロトコルとすることができる。

#### 【0027】

イーサネット・ドライバ362およびADSLドライバ364などのレガシイ装置ドライバ362をストリーミングするために、ゲートウェイはレガシイ・スタック352を提供する。レガシイ・スタック352はTCP/IPスタック3

52およびSPX/IPXスタック354などのOSIプロトコル・スタックをサポートする。レガシイ・スタック352はストリーミング・レガシイ装置ドライバ362および経路選択ハンドラ355と通信する。経路選択ハンドラ355はストリーミング・レガシイ装置ドライバ362と、レガシイ・スタック352と、ペイロード/プロトコル・プロセッサ312と通信する。レガシイ・スタック352はレガシイ・アプリケーション350とも通信する。

#### 【0028】

ゲートウェイ300は各種のネットワークの間の接続点として機能して、イーサネット、デジタル交換線(DSLおよびADSL)、電源線交換機(PLX)、X-10、およびCEBusを含むが、それには限定されない、制御ネットワークおよびデータ・ネットワークのサポートを行う。

#### 【0029】

ゲートウェイ300は、ノード管理およびデータ経路選択の2つの主機能を有する。ノード・マネジャーとして、ゲートウェイ300はネットワーク・ノード状態情報の発見と、列挙と、検索と、保存と、処理との責任を負う。その状態情報はノード・データベース308に保存される。ノード・データベース308は汎用共通アプリケーション言語(CAL)仕様を基にしている。EIA-600で定められたCEBusはバス装置を制御するための工業規格制御言語である。EIA-600は家庭LAN内で使用するための共通アプリケーション言語用の骨格を提供する。汎用CALはEIA-721シリーズの規格(EIA-721.1、EIA-721.2、EIA-721.3、EIA-721.4を含む)で定められる。CEBus工業協議会(CIC)は、汎用CALを使用するための「文法」規則を定めることによってその骨格に肉付けするホーム・プラグ&プレイ(HPP)を定めている。

#### 【0030】

HPP仕様は、家庭内の製品およびシステムが家庭の状態に応じた動作を行えるようにする、それらの製品やシステムのための1組のハードウェア特性の詳細を定めている。たとえば、その仕様は、「住人が居ない」または「住人は在宅しているが寝ている」などの家庭内の種々の状況を識別して、安全装置を作動状態

にする、屋内灯を消す、または温度設定に類似する適切な動作を家庭システムが行えるようにする。HPP仕様は家庭制御用のウィンドウズ95PCベース・アプリケーションを開発するための情報も含む。

#### 【0031】

EIA-600内で定められた共通アプリケーション言語は、種々の産業分野（たとえば、娯楽、コンピュータ、暖冷房、台所用品、等）で製造された家庭LAN製品の間の通信のための枠組を提供する。各産業分野は、その製品がその言語を使用する基準を成す「アプリケーション・コンテキスト」（すなわち、文法規則）を定める。CICは、種々の産業分野が「調和の取れた」アプリケーション・コンテキストを開発することを支援する支持組織として活動するために設立された。CICのHPPは、相互に運用できるCALをベースとする製品で家庭LAN市場を求めているそれらの産業分野のための調和の取れたアプリケーション・コンテキストの概要である。

#### 【0032】

CEBus/汎用CAL仕様の全体を参照することによりその仕様はここに含まれる。

#### 【0033】

ゲートウェイ300は、ノード状態変化を監視し、ユニバーサル規則定義言語(RDL)シンタックスを用いて定義された規則に従ってそれらの状態変化に対して動作する規則エンジン314を提供もする。

#### 【0034】

ゲートウェイ300は、ディスパッチ制御ブロック(DCBs)を用いることによってデータ・ストリーミング・タスクと経路選択タスクの取扱いもする。経路選択ハンドラ355はPLXを通じるTCP/IPトンネリング、プロキシ・サーバ、およびネットワーク・アドレス翻訳のような機能を行う。シム(328、330)は、非PLX電源線をベースとするノードをPLXネットワークに継目なしに挿入できるようにするために設けられる。ゲートウェイ・ファウンデーション・クラス304は、ユーザー・インタフェースおよびシステム管理アプリケーション302が使用するオブジェクト指向APIライブラリを提供する。



ゲートウェイ300は、レガシイ、非ユニバーサル・ネットワークAPIs (MFC、TLI、ソケット、家庭API、等) を使用するアプリケーションがネットワーク・ノードと通信できるようにするサポートを提供する。これによりネットワークを通じてのレガシイ・ネットワーク・データの透明なデータ・ストリーミングを行うことができるようにされる (イーサネット、ADSL、等)。それには、ジャバおよびウェブブラウザを用いてゲートウェイ300により管理されるノードを制御する能力が含まれる。

#### 【0035】

オペランド・データベース308はゲートウェイ300により保存されているノード・データの保存場所である。ノード・データは各クライアント・ノード、事象待ち行列、クライアント・ノード・バインディングス (bindings) および規則から得られたノード構成輪郭を含む。ゲートウェイ300のその他の1組のアクセス規則306によってノード・データベース308をアクセスする。

#### 【0036】

事象ハンドラ310と規則エンジン314はクライアント・ノードで起きる状態変化の結果として起きる行動の責任を負う。規則エンジン314は状態変化に関連する規則を解釈し、かつ規則評価の結果としてトリガされることがあるスケジューリング通知要求またはCAL送り要求の責任を負う。規則エンジン314はノード・データベース308および事象ハンドラ310と一緒に動作する。事象ハンドラ310は事象待ち行列を処理し、事象通知を実行する。それらの事象通知はゲートウェイ300の他のコンポーネントへの内部通知と、事象通知を受信することを登録したアプリケーション302への外部通知とを含む。ゲートウェイ300とアプリケーション302との間の相互作用はゲートウェイ・ファウンデーション・クラス304を介して行われる。

#### 【0037】

ディスパッチャ320は、送/受待ち行列の取扱いと、スレッド (thread) 初期化と、管理等を含む、ゲートウェイ300の実行時間オペレーションを指示する責任を負う。管理を要求するゲートウェイ活動はアプリケーションCA

L送り要求タスクと、生ストリーミング・データ要求タスクと、事象受信タスクと、バックグラウンド・ハウスキーピング・タスクとを含む。ディスパッチャ314は汎用インタフェースをネットワーク装置ドライバ322、326に提供して、ドライバ322、326によりサービスされる下部のネットワーク制御器ハードウェアについての詳細な知識を必要とすることなく各種のネットワーク・ノードのサポートを行えるようにする。

#### 【0038】

デバイスドライバ322、325はネットワーク通信ハードウェア(USBまたは並列ポートPLXノード、X-10シム、CEBusシム等)と直接通信し、かつデータ・リンク層20の機能(たとえば、MACヘッダーの発生/解釈、低レベル・タイミング等)を取り扱う。ドライバ322、326のうち、下部ハードウェアと直接話し合う部分は、ドライバとハードウェアとの間に存在し得る種々の接続(たとえば、ユニバーサル直列バス(USB)、並列ポート、ファイヤ(Fire)ワイヤ・ポート、PCIスロット、ISAスロット、直列ポート等)をサポートするために独自のプラットフォームであるのが普通である。ゲートウェイ300は、レガシイX-10およびCEBusノードから受けとった制御データの、ゲートウェイ300により用いられるフォーマットへの変換を取り扱うためにシム・コンポーネント328、330を提供する。

#### 【0039】

ペイロード/プロトコル・プロセッサ312内のペイロード/プロトコル・ハンドラ316-318は受信されたパケット・データの構文解析および処理の責任を負う。データ・ハンドラ316-317は経路選択ハンドラ355と一緒に動作する。経路選択ハンドラ355は生データ・パケット・ヘッダを構文解析する。ルー・ハンドラ355は、種々のレガシイ・プロトコル・スタック352とレガシイ・ストリーミング・ドライバ362の間のデータの流れの指示を助けるアドレス表および経路選択表を調べもする。

#### 【0040】

種々のプラットフォームおよび種々のオペレーティングシステムを横切るコード移植性(portability)を維持するために、プラットフォームに特

有のコードが、RTOS 311により提供されるプラットフォーム分離 (abstraction) 層内で分離される。典型的な実行時間サービスはスレッド・スケジューリングと、メモリ管理と、割込み処理と、クロックと、同期化と、優先順位スケジューリングと、初期化等を含む。

#### 【0041】

ゲートウェイ300内のコンポーネントの多くは、ノード・データベース308と直接にまたは間接に相互作用する。データベースを最初に構築し、維持するタスクはディスパッチャ320と、ドライバ322、326と、ペイロード/パケット・ハンドラ312とによって主として取り扱われる。ユーザー・インタフェースおよび管理アプリケーションがゲートウェイ・ファウンデーション・クラス304を介してノード・データベース308をアクセスおよび/または変更する。

#### 【0042】

ノード・データベース308は、実時間アクセスが速くて効率的であるように構成されている。速さ/メモリ使用の間の妥協が用いられ、アクセス速度よりも効率的なメモリ使用に優先順位が通常与えられる。ある実施形態では、ノード・データベース308の部分が不揮発性記憶装置に保存される。

#### 【0043】

ノード・データベース308はクライアント・ノード構成とノード状態情報を含む。これはコンテキストと、オブジェクトと、各クライアント・ノードに関連するインスタンス変数 (IV) 情報を含む。この情報はノード発見中にネットワーク・コマンドとネットワーク要求を用いて得られる。この情報のうち、ダイナミックIVsが固定記憶装置に保存される。ほとんどのオブジェクト (ノード制御またはコンテキスト制御は含んでいない) に対して、これはノードの状態を定める現在の値IVsを含む。

#### 【0044】

ノード・データベース308はゲートウェイに特有のある情報をも含む。各ノードに対して存在するノード・コンテキスト以外に、ノード・データベース308は、ノード管理のために必要とされるオブジェクトを含んでいるデータベース

・サーバ・コンテキストを含む。このデータベース・サーバ・コンテキストは、ゲートウェイ300によって知られているクライアント・ノード・アドレスのアドレスを含むネットワーク・ノード・リスト・オブジェクトを含む。

#### 【0045】

ノード・データベース308はアクセス法の汎用セットを通じてアクセスされる。ほとんどのアクセスはネットワーク・メッセージを通じて直接または間接に来るので、データベース・アクセス法306は典型的なネットワーク獲得およびセット法を用いて変数値を獲得およびセットする責任を負う。それらの方法には GetValue、GetArray、SetValue、SetArray、SetOn、SetOffが含まれる。どの方法が適切であるかは取り扱われているインスタント変数のタイプに依存する。たとえば、CALネットワークは、ブーリアン（GetValue、SetValue、SetOn、SetOff）、数値（GetValue、SetValue）、キャラクタ・ストリング（GetValue、SetValue）、およびデータ（GetArray、SetArray）の4つのIVデータ・タイプを定める。

#### 【0046】

ほとんどの部分に対して、それらの方法は入力キーの類似のセットを取り、それらは取り扱われているデータと、任意の数の入力引数を識別する。その後で各方法は求められた情報（もしあれば）と状態を戻す。たとえば、CAL IVsを取り扱う方法の汎用プロトタイプは下記のものに類似する。

```
ENUM STATUS<method>(NodeID, Context, ObjectID, IVID, args, .... ★returnVal)
```

この場合には、NodeID、ContextID、ObjectID、およびIVIDはキーである。非CALオブジェクトに対しては異なるキーセットが用いられる。

#### 【0047】

インスタンス変数にセット操作が起きると、事象通知が事象ハンドラ310を通じて規則エンジン314に対して行われて、状態変件事象が起きたことを通知する。その後で規則エンジン314は、変化したインスタンス変数に属する規則状態／報告状態を解釈し、行動が保証されているならば、事象ハンドラ310に

よって事象のスケジュールを定める。この結果としてディスパッチャ320を迂ってネットワーク送り要求が発生し、またはファウンデーション・クラス304を通じてより高いレベルのアプリケーション302に通知されることになる。

#### 【0048】

上記のように、ノードIVを獲得またはセットするためにアクセス呼び出しが行われると、4つの情報（またはキー）が通常提供される。それらにはNode IDと、Context IDと、Object IDと、IV IDとが含まれる。

#### 【0049】

Node IDは構成中に割り当てられた（これは製造時に、または実行時間中に動的に、生ずることがある）ノード・アドレスである。PLXネットワークの場合には、これは4バイト・フィールドであって、0x00000001と0xffffffffの間値を取ることができ、この範囲より上のアドレスは放送またはその他のPLXに独自の使用のために取って置かれる。非PLXネットワークの場合には、同じアドレス・マッピング／翻訳が通常求められる。経路選択ハンドラ355はアドレス翻訳問題に気を付ける。

#### 【0050】

Context IDは2バイト長であって、大きなバイトは選択可能なコンテキスト数で、その値は0xA0・0xDEの範囲であるが、使用されなければ0である。小さいバイトは0x00・0x9Eの範囲内のコンテキスト・クラス値である（それらは汎用CAL仕様で定められる）。

#### 【0051】

Object IDは0xD1・0x3Eの範囲の1バイトオブジェクト番号である。

#### 【0052】

IV IDは所与のコンテキストおよびオブジェクト・タイプ内のインスタンス変数を特定するasciiキャラクタ（または複数のキャラクタ）である。

#### 【0053】

ノードはコンテキストのグループにより記述される。各コンテキストはオブジェクトのグループにより構成されている。各オブジェクトはIVsのグループに

より構成されている。データベースは、ちょうど論じている4つの情報について、IVに対するアクセスが下記のルックアップ・アルゴリズムを含むように構成されている。

【0054】

1) Node IDは各ノードはコンテキスト・レコードのアレイを示す。ノード・リスト・エントリにマップする（通常はハッシング・アルゴリズムを用いて）。Context IDは、所望のコンテキスト・レコードを獲得するための直接インデックスとして使用できるように構成されていないので、ここでは直線的なルックアップが起きる。ノードはほんの少し（2または3）のコンテキストを通常有する。各コンテキスト・レコードはオブジェクト・レコードのアレイ（またはリンクされたリスト）に対するポインタを含む。

【0055】

2) Object IDは、所望のオブジェクト・レコードを獲得するための直接インデックスとして使用できるようなフォーマットのものである。各オブジェクト・レコードはIVレコードのアレイに対するポインタを含む。

【0056】

3) IV IDを基にして動作する所望のIVを探すために直線サーチが用いられる。これは、ほとんどのクライアント・ノードに対してほんの僅かなIVsがノード・データベース308に保存されるから効率的である。ほとんどのGET要求/SET要求はIVの現在の値に関する。現在の値がIVリストに常に保存されていると、集中的な直線サーチが必要とされることはまれである。

【0057】

ネットワーク・ノード・リストを住ませる（to populate）ためにゲートウェイ300により用いられる発見プロセスは、ノードのタイプに依存して変化する。CALを用いる典型的なPLXに対しては、下記のアルゴリズムが用いられる。

【0058】

1) CAL「Ping」要求がネットワーク上で放送される（これは初期化中に通常行われるが、定期的にも同様に行うことができる）。この要求のフォー

マットは他の任意のCALコマンド／要求のフォーマットに類似し、下記のパラメータを用いる。

**【0059】**

**【数1】**

**Context Class = 0 (Universal Context),**

**Object Number = 1 (Node Control Object),**

**Method = 50h = PING\_REQUEST (not defined in the generic CAL spec.),**

**IV = <any IV OK>.**

この要求は、ノードへ直接送ることができる点、または全てのノードへ放送できる点で、標準CAL要求とは異なる。

**【0060】**

2) CAL Ping要求を受信するノードは送り出しノードへCAL Ping要求を送り返す。このパケットは「応答」と呼ばれているが、パケットのシンタックスは従来のCAL要求パケットとは異なる。その応答は、ping要求を放送として送ることができるようにするために異なるのである。その応答パケットのフォーマット（実際にはCAL応答ではなくてCALコマンド／要求に類似する）は、方法コードが50hの代わりに51hであることを除き、ping要求に類似する。

**【0061】**

3) ゲートウェイ300が、ネットワーク・ノード・リストに掲載されていないノードからping要求を受けると、ゲートウェイ300はそのノードをリストに加え、ノードのIVs情報をノード・データベース308に付加する。

**【0062】**

4) 希望によっては、ゲートウェイ300は、ある長さの時間聞かれなかったノード・リスト内の任意のノードにCAL要求を送る。そのノードが応答しなければ、それはそのノードから除去される。

**【0063】**

5) ping応答パケットは、クライアント・ノードに電源が投入された時

またはリセットされた時にも送られる。

#### 【0064】

レガシイ制御ノード（X-10ノードまたはCEBusノードなど）では、ノード発見プロセスは、レガシイ・アプリケーションに従って変化する。ゲートウェイ300はシム328、330の支援でそれらのノードを処理する。シム328、330は、上記CALノード発見法と、非PLXドライバにより使用されるレガシイ法を知っている。ping要求に応答して、シムはping要求を、レガシイ制御ネットワークでノード発見を実行するために求められる同等の要求（または要求セット）に変換する。ノードが発見されると、シム328、330はping応答パケットを発生して、それらをゲートウェイ300の上側の層へ送る。ゲートウェイ300の残りに関する限りは、それらのレガシイ制御ノードはPLXノードである。シム328、330については以下でさらに説明する。

#### 【0065】

ゲートウェイ300によってユーザー・インタフェースおよび管理アプリケーション302がファウンデーション・クラス304を通じてノード・データベース308をアクセスできる。アプリケーション302によるノード・データベース308のアクセス性を高めるために、ゲートウェイ300は多数のアプリケーション・サーバの存在をサポートする。ここに、1つのサーバがアクティブアプリケーション・サーバとして動作し、他のサーバが待機アプリケーション・サーバとして動作する。

#### 【0066】

各サーバは同じノード・データベース308を保存し、ネットワーク・トラフィックを聴取し、ネットワーク・ノード状態変化を監視し、かつそれに従ってそのノード・データベース308を更新することにより、他のノードと同じ日（同期させられる）まで、そのノード・データベース308情報を保持する。しかし、アクティブアプリケーション・サーバは、状態変化に関連している諸規則を実際に評価し、求められている事象通知を実行するノードのみである。アクティブサーバがなんらかの理由で動作しなくなったとすると、待機サーバがこれを検出して「アクティブにされる」ことになる。アプリケーション・サーバ・ノ



ド402に対するアクティブプロセスが、同期化ブロック404で始まる、図4に示されている。その同期化ブロックではノード402はそれのノード・データベース308を更新する。ノード・データベース308を更新した後で、プロセスは判定ブロック406へ進む。この判定ブロック406で、ノードがアクティブサーバ状態にあると、プロセスはアクティブサーバ・ブロック408へ進み、さもなければ、プロセスは待機サーバ・ブロック420へ進む。アクティブサーバ・ブロック408が終了すると、プロセスは判定ブロック412へ進む。この判定ブロック412で、サーバ睡眠要求が検出されると、プロセスはセット待機ブロック410へ進み、さもなければプロセスは判定ブロック414へ進む。この判定ブロック414で、クライアント要求が検出されると、プロセスは応答ブロック418へ進み、さもなければプロセスは同期化ブロック404へ戻る。セット待機ブロック410または応答ブロック418が終了すると、プロセスは同期化ブロック404へ戻る。

#### 【0067】

待機サーバ・ブロック420が終了すると、プロセスは判定ブロック422へ進む。この判定ブロック422で、確認応答されなかったクライアントが検出されると、プロセスはアクティブセット・ブロック424へ進み、さもなければプロセスは同期化ブロック404へ戻る。アクティブセット・ブロック424が終了すると、プロセスは通知ブロック426へ進む。このブロックでは、現在のサーバ・ノードがアクティブ状態になりつつあることを他のアプリケーション・サーバ可能ノードに通知する。ブロック426が終了すると、プロセスは同期化ブロック404へ戻る。

#### 【0068】

クライアント・ノードで状態変化が起きた時に、規則エンジン314と事象ハンドラ310が含まれることになる。SET方法が実行された時にそのような状態変化が起きるので、各SET方法はその変化を事象ハンドラ310を介して規則エンジン314に通知する。その後で規則エンジン314は、変化したIVに関連している任意のノードの存在を調べ、なんらかの事象通知が必要とされているかどうかをそれは判定する。ユーザーが、ファウンデーション・クラス304

を用いてユーザー・インタフェースを介して種々のノードを結び付けると、ユーザーは規則を作成できる。あるノードと、明示のユーザー定義を必要としない従来のオペレーションに対してデフォルト・ノードは存在することもできる。

#### 【0069】

規則は簡単にもできれば、複雑にもできる。簡単な規則は、クライアント・ノードの間の1対1または1対多数の結合関係の特徴するものである。簡単な1対1規則の例は、「灯火スイッチAがオンにされると、電球Bが点灯する」ということに類似するようなものと読めるであろう。1対多数規則の例は「灯火スイッチAがオンにされると、電球A、B、C、Dが点灯する」と読めるであろう。複雑な規則は多数対1または多数対多数の結び付きを取り扱う（たとえば、「灯火スイッチAがオンにされ、かつユーザーが安全クリアランス（clearance）有すならば、電球A、B、C、Dが点灯する」）。

#### 【0070】

ゲートウェイ308は、ユニバーサル規則制定言語（RDL）として知られている規則制定のための多用途シンタックスを提供する。図5は規則の構成要素と、アプリケーション502と規則504との間の相互作用とを示す構造図である。図5に示すように、アプリケーション502は、規則504を含めた、1つまたは複数の規則を制定する。規則504は、条件付き表現506を含む「if」文を含むことができる。条件付き表現506は1つまたは複数の定数508と、1つまたは複数の演算子512と、1つまたは複数のIvsとを含むことができる。規則504内の「then」クローズはアプリケーション502に返される通知514を特定する。各定数は値を含む。各演算子は演算子識別名（ID）を含む。各IVはIDと、値と、トリガ状態とを含む。トリガ状態は縁部をトリガされたものと、非縁部をトリガされたものを含む。各通知は通知IDとメッセージIDを含む。

#### 【0071】

規則は事象と行動を含む。行動は、アプリケーションに特有であるストリングのアレイである。事象は、所与の状態がネットワークに存在しているか否を記述するブール表現である。事象ストリングは下記の文法を用いて指定される。その

文法は定義されていないターミナル記号 `integer_constant`、`character_constant`、`floating_constant`、および `string` を有する。識別子ターミナルは特定のフォーマットを有することに注目されたい。脱字記号「```」ポンド符号「`#`」の後の最初の `integer_constant` は16進法で指定されたノード・アドレスである。2番目（任意選択）の `integer_constant` は16進法でのCALコンテキスト数である。3番目の `integer_constant` は16進法でのCALコンテキスト・クラスである。4番目の `integer_constant` は16進法でのCALオブジェクト数である。`character_constant` はCALインスタンス変数（IV）である。識別子の前の脱字記号は「縁部をトリガされた」`iv`と「非縁部をトリガされた」`iv`を指定する。ポンド符号は、`iv`が「レベル」または「既存値」であることを意味する。規則は1つの「縁部をトリガされた」識別子を有するのみである。

【0072】

規則文法

【0073】

【数2】

**IF OR expression THEN ACTION expression****OR expression**

*AND\_expression*  
*OR\_expression* || *AND\_expression*

**AND expression**

*equality\_expression*  
*AND\_expression* && *equality\_expression*

**equality\_expression**

*relational\_expression*  
*equality\_expression* == *relational\_expression*  
*equality\_expression* != *relational\_expression*

**relational\_expression:**

*additive\_expression*  
  
*relational\_expression* < *additive\_expression*  
*relational\_expression* > *additive\_expression*  
*relational\_expression* <= *additive\_expression*  
*relational\_expression* >= *additive\_expression*

**additive\_expression:**

*multiplicative\_expression*  
*additive\_expression* + *multiplicative\_expression*  
*additive\_expression* - *multiplicative\_expression*

**multiplicative\_expression:**

*primary\_expression*  
*multiplicative\_expression* \* *primary\_expression*  
*multiplicative\_expression* / *primary\_expression*  
*multiplicative\_expression* % *primary\_expression*

**primary\_expression:**

*identifier*  
*constant*  
*string*  
(*OR\_expression*)

*identifier*: \**integer\_constant*.*integer\_constant*.\**integer\_constant*.*integer\_constant*.*character\_constant*  
#*integer\_constant*.*integer\_constant*.\**integer\_constant*.*integer\_constant*.*character\_constant*

**constant:**

*integer\_constant*  
*character\_constant*  
*floating\_constant*

**ACTION expression**

*Notify\_applications\_registered\_for\_this\_rule*  
*Send\_PLX\_packet\_to\_node*

あるレガシイ・シンタックス (たとえば、RDL以外のシンタックス) を用い

る規則が埋め込まれている非PLXノードに対しては、規則エンジン314は（シム328、330からの助けて）レガシイ・シンタックスをRDLシンタックスに変換する。

#### 【0074】

データベース320は、ゲートウェイ300の内部で起きるオペレーションの順序および流れを指令する責任を負う。初期化中は、ディスパッチャ320はRTOS311を呼び出して、実行時間オペレーションを管理する種々のハンドラ・スレッドをつくる。ハンドラ・スレッドは、1) 受信ペイロード・スレッド、2) 生データ（ストリーミング）送信スレッド、3) CAL送信スレッド、および4) 低い優先順位のアイドル・スレッド（任意）、を含む。それらのスレッドは優先順位の順に記載した。

#### 【0075】

上記ハンドラ・スレッドのほとんどはパケットをクライアント・ノードへ送る。デバイスドライバ送りルーチンの呼び出しを行う前に、それらのスレッドはドライバが送り要求を取り扱えることを示すTxFreeCountセマフォでまず待つ（睡眠）。また、CAL送信スレッドは、「遅延された」モードにあるクライアントへはCALパケットが送られないようにする。遅延されたノードについては以下で説明する。

#### 【0076】

ディスパッチャ320は、ハンドラ・スレッドが働き掛ける待ち行列の管理も行う。ハンドラ・スレッドおよびゲートウェイ300のその他の構成要素はディスパッチャ320を呼び出してパケットをその待ち行列に加え、かつその待ち行列からパケットを除去する。ディスパッチャ320はディスパッチ制御ブロック（DCBs）を用いてCAL要求パケット／CAL応答パケットおよび生データ流部分を記述する。

#### 【0077】

DCB構造が（C／C++シンタックス）で下記のように定められる。

#### 【0078】

【数3】

```

typedef struct sDCB
{
    void          *link;
    UINT8 *buffer;
    UINT32      destDevAddress;
    UINT32      srcDevAddress;
    UINT16      timeStamp;
    UINT16      reserved1;
    UINT8       destSocket;
    UINT8       srcSocket;
    UINT8       sequenceNo;
    UINT8       bufferSize;
    UINT8       controlInfo;
    UINT8 reserved2[5];
}
tDCB, *PDCB;

```

linkフィールドはどのような目的に対してもDCBの所有者により使用できる。bufferフィールドおよびbufferSizeフィールドは、DCBにより記述されているパケット／断片データを指し、そのサイズを定める。destDevAddressおよびsrcDevAddressは、DCBに関連する宛先ノードと送り手ノードを特定する。destSocketおよびsrcSocketはノード内の宛先アプリケーション／送り手アプリケーションをさらに特定するために使用することもできる。timeStampフィールドは、種々の時間切れ条件の処理を支援するためにゲートウェイ300により使用される。sequenceNoは、主として生データ・パケットストリームのためのパケットに命令するために用いられる。controlInfoフィールドは、DCBおよび関連するデータ・パケットの求められた特殊な特徴／処理を指示するために用いられる種々のビット・フィールドおよびフラグを保存するために用いられる。それらのビットはデータ暗号化、認証、およびデータ・ストリーミングのような諸特徴を制御する。reserved1フィールドおよびreserved2フィールドはゲートウェイ300により内部で用いられ、かつ外部アプリケーションにより用いられる。

#### 【0079】

受信ペイロード・スレッドは通常、ディスパッチャ320によりスタートさせられた最高優先順位のスレッドであって、デバイスドライバ322、325が受

信パケットを行列にした時に常に実行する。デバイスドライバ322、325における割り込み／ポール・サービス・ルーチンはこのスレッドを提供できる。このスレッドはパケットを受信待ち行列から除去し、それらのパケットを更に処理するためにペイロード／プロトコル・ハンドラ312にそれらのパケットを渡す。このスレッドは受信待ち行列に何も残らなくなるまで実行する（またはおそらくあるしきい値に達するまで）実行し、その後でより多くの受信が行列にされるまで休止する。

#### 【0080】

生データ送信スレッドは2つの条件が真である、すなわち1) 生データ／ストリーミング送信要求がアプリケーションによって待ち行列にされた、および2) 任意のより高い優先順位のスレッド／割り込みハンドラが実行を終了した時に実行する。生送信スレッドは送り待ち行列中の次のエントリを獲得し、それを適切なデバイスドライバ送りルーチンに渡す。これは、送り待ち行列内にもう報告が残っていないまで、またはあるしきい値に達するまで、繰り返される。その後で生データ送信スレッドは新しい送信の予定が組まれるまで休止する。

#### 【0081】

CAL送信スレッドは生データスレッドに類似する。その2つの間の主な違いは、CAL送信スレッドが種々の送り待ち行列を取り扱うことである。CAL送信スレッドはCAL要求事象に対して働き掛ける。CAL要求事象はアプリケーション304から来る要求の結果として予定が通常組まれる。それらのアプリケーション要求はディスパッチャ320によりCAL送り待ち行列に置かれる。遠方のノードまたはノード・データベース308に保存されている状態情報を読み出し（獲得）および書込む（セット）するためにCAL送信を使用できる。規則エンジン314により解釈される規則中に定められている事象をトリガする内部状態変化および外部状態変化によってCAL送信を発生することもできる。

#### 【0082】

システム内で低い優先順位でアイドル・スレッドが実行するが、それを行うかどうかは任意である。もし使用されると、アイドル・スレッドは、より高い優先順位のスレッドが実行している間に安全に無期限に延期できる低い優先順位のタ

スクを取り扱う。低い優先順位のタスクは次の行動をすなわち。1) メモリ管理タスクおよびがらくた収集タスク、2) 非応答ノードを検出および古びさせるために見張りパケット／pingパケットを送り出す、3) キャッシュされたデータベース情報を非キャッシュされた（持続する）予備保存コピーに同期させる。アイドル・スレッドが支持されると、アイドル・スレッドにより呼び出されるべき低い優先順位のタスクを他のモジュールがスケジュールできるように、ディスパッチャ320は他のモジュールにインタフェースを提供する。

#### 【0083】

デバイスドライバ322、326はディスパッチャ320（ディスパッチャ制御ブロックすなわちDCBにより記述されている）から汎用要求を受け取り、それらの要求を基礎をなすネットワーク・ノードに適切なI/O要求に変換する。

#### 【0084】

デバイスドライバ322、326はネットワーク・ハードウェアから受取ったパケットも処理し、以後の処理のためにその後で送られるDCBsを組み立てる。ディスパッチャ・スレッドはゲートウェイ300のうち、デバイスドライバ322、326にインタフェースする唯一のコンポーネントである。デバイスドライバ322、326によりディスパッチャ320に与えられたインタフェースはDCBを通されるDriverSendインタフェースを含む。媒体アブストラクション・コンポーネント（MAC）モジュールはMACヘッダを組み立て、ネットワーク制御器ハードウェアへの実際のパケット送りを開始する。ドライバ割り込み／ポーリング・ルーチン待ち行列はペイロード受信待ち行列における事象を受け取り、その後の時刻にその待ち行列は受信ペイロード・スレッドによって処理される。

#### 【0085】

以下の説明のほとんどは、デバイスドライバがPLXをベースとするノードをドライブしていると仮定している。しかし、上側の層に提供されるインタフェースは、他のドライバがゲートウェイによりサポートされるように汎用である。シム328、330は、X-10ノードおよびネイティブCEBusノードなどの非PLXベースとするノードが、ゲートウェイ300により継ぎ目なしにサポー



トされるようにするために設けられている。

#### 【0086】

たとえば、PLX制御ドライバ323などのドライバがロードされると、ドライバ初期化モジュールが、そのドライバによりサービスされるネットワーク・インタフェース・ハードウェアを機能的な状態に置く。初期化が終わると、ドライバ323とハードウェアがパケットの送受信できる状態にある。ドライバ初期化の一部が、受信ハンドラISR／ポールルーチンの設定と、ハードウェア資源の決定／保留と、任意のハウスキーピング・スレッドのスタートとを通常含む。通常、送信時間切れと死んだ装置ドライバ問題／死んだハードウェア問題を調べるためにタイマ・スレッドが用いられる。

#### 【0087】

送り時には、ディスパッチャ320はパケット／断片データ・アドレスと、制御情報をDCBを用いてドライバ322に供給する。その後で、ドライバ320は適切なMACヘッダをつくり、送信を開始する。同時に取り扱える送信の数は、初期化中に読出すことができるハードウェア／ファームウェアに依存する値（TxFreeCountと呼ばれている）である。

#### 【0088】

送信データをネットワーク制御器にコピーした後で、DriverSendルーチンが送り時間切れタイマを開始する。この点で、送信終了を示す応答が同期して受信されないと（DriverSendルーチンが戻る前の送りに続く）、ネットワーク・ハードウェアが送信終了状態を示すまで送信は「係属」状態にある。送信が係属状態にある間は、ネットワーク制御器が利用できるバッファスペースを有する限り、TxFreeCountにより定められるように、より多くの送りが起きることができる。係属している送りの数がTxFreeCountにひとたび等しいと、DriverSendルーチンはロックされ（それ以上の送りは起きないことを意味する）、ISR／ポール・ルーチンが送信成功状態または送信時間切れ状態の受取りに続いてそのロックを解除するまでロックされたままである。

#### 【0089】

ドライバ割り込み／ポール・サービス・ルーチン（またはISRハンドラ）ルーチンはシステム内で最高の優先順位で実行し（割り込みコンテキストが好ましい）、受け取られたパケットのMACヘッダの解釈／除去の責任を負い、後で適切なペイロード・ハンドラにより処理するためにディスパッチャ320でパケットを待ち行列化する。受信パケットは次の部類、1）内部（またはローカル）制御／状態パケット、2）CALコマンド／要求、3）CAL迅速応答、4）CAL遅延応答、または5）生ストリーム・データの1つに入る。

#### 【0090】

内部制御／状態パケットは、送信終了または誤り状態を示すために主として用いられる。ISRハンドラにおける送信状態パケットの処理は、状態が適用される送信パケットのタイプ（たとえば、CAL／生、ローカル／遠隔）にいくらか依存して通常変化する。

#### 【0091】

生断片CALパケットとCALパケットの両方に対して、送信が成功裡に終了させられたことを状態が示すと、ISRルーチンはTxFreeCountを調整して次の送りが起きることを可能とする。時間切れ誤りの場合には、CAL送信は、送りが成功するまで適切な何回か再試行するか、ある最大再試行回数に達した後で放棄することが好ましい。

#### 【0092】

生データ断片の送信時間切れは、ある場合に（生データストリームの性質と、受信側で予測されるものとに依存する）、追加の再試行処理を要求することができることを除き、CAL送信に類似するやり方で取り扱われる。生ストリーム断片は、データストリームの管理の支援のためにパケット・シーケンシングを通常用いる。生ストリーム受信アプリケーションの要件に依存して、時間切れ取扱いコードが、その時間切れを送信するばかりでなく、係属中の追加の生送信（より大きい一連番号を盛っているもの）が成功に終わるか否かとは無関係に、それらの係属中の送信のいずれも再試行することが望ましいことがある。これは受信アプリケーションまたは経路選択ハンドラが昇順の順序でパケットを確実に受け取ることを支援する。受信app／経路選択装置はそれらを落とすことにより狂っ

た順序を取り扱う必要がある。

【0093】

CAL要求／応答および生ストリームパケットはISRハンドラにより待ち行列にされ、受信ペイロード・スレッドおよび適切なペイロード・ハンドラにより後で処理される。

【0094】

CAL応答が係属中であるような遠隔ノードへ新しいCAL要求が送られないことを検査するために、特殊な取扱いもCALパケットで用いられる。また、CAL要求が遠隔のノードから受け取られるとすると、通常は、適切なCAL応答が送られるまで新しいCALパケットはそのノードに送られることを許されない。

【0095】

レガシイ制御ネットワーク装置ドライバ（X-10ドライバ328およびCE Busドライバ332など）に対しては、シム323、330は、それらの非PLXドライバがゲートウェイ300と動作できるようにするインタフェースを提供する。送信時には、シム323、330はディスパッチャ320により送られるCALパケットを調べ、それらのパケットを、基礎を成しているレガシイ・ドライバ／装置により認識される同等のフォーマットに変換する。受信時には、シム323、330は受けたレガシイ・データをCALフォーマットに変換し、DCBを組み立て、そのDCBをディスパッチャ320まで送る。

【0096】

パケットの送受信中はシム323、330は、4バイトDCBアドレスを、基礎を成すレガシイ・ドライバにより用いられているアドレス・フォーマットに変換することにより、あるアドレス変換を通常実行する。

【0097】

受け取られたパケットはCAL制御メッセージまたは生ストリームデータ（それはプリンタ・データ、トンネルを通るように送られた（tunneled）イーサネット／ADSLデータ等を含むかもしれない）を制御する。それらのペイロードの各々は、ペイロード／プロトコル・プロセッサ312により提供され

る適切なペイロード／プロトコル・ハンドラにより取り扱われる。CALハンドラ318は、制御ネットワーク・ノードから受け取られたCAL要求／応答制御 packets を構文解析および処理するCALインタプリタを含む。ペイロード・ハンドラ・スレッドはCAL packets を更に処理するためにクライアントからCALハンドラへ送る。CALインタプリタはある基本CAL構文解析ルーチンを、CALメッセージを取り扱う他のゲートウェイ・コンポーネントにも提供する。

#### 【0098】

ディスパッチャ320がCALハンドラ318を呼び出すと、それは、特に、CALメッセージに対するポインタと、その packets を生じたクライアントのアドレスと、働き掛けるべきクライアントのアドレスとを送る。それら2つのアドレスは、アプリケーション302から packets が生じた場合を除き、通常は同じである。

#### 【0099】

そのCALメッセージはコマンド（要求とも一般に呼ばれる）または応答メッセージである。クライアント・ノードからのCALコマンドは通常は、「私の状態変数にある新しい値にセットする」の作用についてゲートウェイ300に何事か告げる状態変化通知である。アプリケーション304により制御されているようなものなどの、特殊機能クライアントもCALコンポーネントを送り、ユーザーの要求を基にして、他のクライアントの状態（インスタンス）変数を獲得、またはセットする。それらの packets は次の節で説明する特殊取扱いを通常要する。

#### 【0100】

単一コマンドCAL packets は次のようなフォーマットのものである。

```
<contextID><object#><method/macroID>
[<IV> [<arguments>]]
```

CALインタプリタはメッセージをそのコンポーネントに分解し、方法識別子を、後でIDおよび引数パラメータで呼び出される適切なデータベースアクセス法にマップする。データベース・アクセス法306は求められたオペレーションを実行し、IVが変化すると、規則エンジン314に通知される。規則エン

ジン314は変化させられたIVに適用されるどのような規則も評価し、保証されるならば、上記のように事象ハンドラ310により事象通知の計画が立てられる。

#### 【0101】

CALコマンドに応答するためにCAL応答パケットはクライアント・ノードによって用いられる。CAL応答は<status token> [<returned data>] の形のものである。ここに、<status token>は応答 (COMPLETED, FALSE, or ERROR) のタイプの1バイト標識であり、<returned>データはコマンド・メッセージの結果として戻された任意のデータである。

#### 【0102】

それらのパケットの1つが受け取られると、それは元のCALコマンド/要求に関連させられる。クライアントが要求に直ちに応答する時はその関連付けは比較的容易である。応答が遅れた時は関連付けは一層複雑である。関連付を可能にするものは、あるクライアントが以前の要求に対する応答を送るまで、クライアントがCAL要求パケットをネットワークへ送らないかもしれない、という事実である。「遅延させられた」モードにあるクライアントへはゲートウェイ300は要求を送らない。これによってゲートウェイ300はクライアントに最後に送られた要求 (またはコマンド) をクライアント毎に保存できるようにされる。クライアント応答が受け取られると、この情報によってCALハンドラ318は、戻された情報ではすべきことを決定できる。

#### 【0103】

生データ・ハンドラ316は非CAL受信データを処理する。ゲートウェイ300は、種々のアプリケーションで生ずることができる多数の、同時データストリームをサポートする。生データ・ハンドラ316が生データストリームの間で識別するやり方はソケット・フィールドを含む。そこでは独自のソケット番号が各生データストリームタイプに関連させられる。生データストリーム例にはイーサネット/ADSLネットワーク・パケットと、プリンタ・データと、音声ストリーム等が含まれる。生データ・ハンドラ316はソケット番号を調べるため

以外の生データの構文解析はほとんど行わない。パケット・データと、関連付けられたDCBはその後で、そのソケット番号に責任を負う経路選択ハンドラ355へ送られる。

#### 【0104】

経路選択ハンドラ355は、ゲートウェイ300がレガシイ・データ・ネットワーク（イーサネット、ADSL、トークンリング等）からのネットワーク・データ・トラフィックを所望のネットワークへ向け直すことを可能にする機能を提供する。たとえば、レガシイ・ネットワークからのネットワーク・データ・トラフィックを、PLX電源線ネットワークへ向け直すためにゲートウェイ300を用いると、1）追加のイーサネット・ケーブル配線の必要なしに、電源線を基にしたノードを含むように従来のイーサネット・ネットワークを拡張できること、2）広帯域データを電源線を通じて送ることができること、3）電源線をベースとするネットワーク・クライアントのために代用サーバ性能を可能にすること、4）レガシイ・プロトコル・スタック（TCP/IP、PX/IPX、NetBEUI等）を電源線を通じてトンネルを通るように送ることができるようにすること、を含むが、それらに限定されるものではない。

#### 【0105】

経路選択ハンドラ355により実行される重要なタスクはアドレス翻訳である。初期化中は、経路選択ハンドラ355は、ゲートウェイ300から発生されたDCBsを送信する際に、srcDevAddressとして用いられている4バイト・アドレスを得る。経路選択ハンドラ355はこのアドレスを、レガシイ・スタックとドライバの少なくとも一方に適切な形に翻訳するので、ハンドラは通信する。

#### 【0106】

以下の節は、DCBsを用いて異なるプロトコル（たとえば、PLXプロトコル）によってレガシイ・ネットワーク・データ・パケットをトンネルを通るようにして送るために、経路選択ハンドラ355により用いられる方法を説明するものである。トンネルを通るようにして送ることは第1のプロトコルからのパケットを第2のプロトコルに対するパケット内側に封じ込めるすなわち包み込むこと

である。包まれたパケットはその後で第2のプロトコルを介してネットワークを通じて送られる。その宛先に到達すると、包まれたパケットはほどかれて第1のプロトコルからの最初のパケットを現す。

#### 【0107】

初期化中には、経路選択ハンドラ355は前記したようにアドレス・マッピング表／翻訳表を設定する。経路選択ハンドラ355は、DCBごとにサポートされる最大パケット／断片サイズに気付いてもある。経路選択ハンドラ355はこの情報をディスパッチャ320から得る。種々の経路選択ハンドラ355はソケット番号を用いることによって特定される。初期化中に独占的に使用またはダイナミックに得るために周知のソケット・アドレスを保留できる。

#### 【0108】

電源線装置のために設計されているレガシイ・スタックまたはドライバからの送り要求を経路選択ハンドラ355が得ると、経路選択ハンドラ355は送りデータをサポートされているDCBデータサイズより大きくない断片にばらばらにする。その後でDCBが作成され、各断片に一連番号が割り当てられる。最初の断片は常に一連番号0で、最後の断片は一連番号セットの高位のセットが割り当てられる。任意に、別のレベルで機能が適切に提供されなければ、経路選択ハンドラ355はチェックサムをパケットに付加できる。その後でそれらのDCBsは、生送り待ち行列で待ち行列化するためにディスパッチャ320に渡される。これは生送りスレッドを覚醒させ、電源線へ送信するためにそれはDCBsを適切な装置ドライバに渡す。

#### 【0109】

装置ドライバによって各送信が終了させられると、Tx状態がDCBにマークされ、そのDCBは経路選択ハンドラ355に戻される。全てのDCBが戻された後で、経路選択ハンドラ355はレガシイ・プロトコルによって求められた送り終了動作を実行する。時間切れ／再試行の問題がディスパッチャ320と、装置ドライバと、多くの場合にはレガシイ・スタックによっても同様に取り扱われているので、経路選択ハンドラ355は追加の再試行動作を実行しないことを選択できる。

## 【0110】

受信取扱いはより複雑にされたビットである。その理由は、経路選択ハンドラ355が多数の送り手から断片を同時に受けることが可能だからである。また、ある場合には、断片を順序なしで受けることがあり、または断片が脱落することがある。経路選択ハンドラ355はそれらの可能性の全てを取り扱うことができる受信モジュールを有する。通常は、初めにDCB／断片が、ペイロード／プロトコル・ハンドラ316により経路選択ハンドラ355の1つ渡されるので、送りアドレスのために留保されている受信待ち行列に置かれる。各DCBが受けられると、受け手は最後の断片列または受信時間切れを調べる。

## 【0111】

全ての断片が受けられて、完全性検査に合格したら、経路選択ハンドラ355は、受信事象を示すためにレガシイ・プロトコル／ドライバにより求められているステップを実行する。これにはパケットの再組み立てと、おそらくは関連する受信データの、レガシイ・モジュール352により提供されたバッファへのコピーが通常含まれる。各受信DCBにより記述された物が処理された後で、DCBはフリーDCBリストに戻される。

## 【0112】

ゲートウェイ・ファウンデーション・クラス304がオブジェクト指向概念（たとえば、Java、C++、smalltalk等）を基にしており、各種のアプリケーションがノード・データベース306、規則エンジン314、事象ハンドラ310、およびゲートウェイ300により提供されるその他のサービスに保存されているノード情報をアクセスおよび管理する方法を提供する。これによってエンドユーザー・アプリケーション302は広範囲の有用な諸特徴をエンドユーザーに提供できる。たとえば、ゲートウェイ・ファウンデーション・クラス304は自立アプリケーションとJava browser appletが、ノード・データベース308において記述されているノードの列挙と、監視と、制御とを可能にする。それらのアプリケーション／appletsは、規則定義言語を用いて簡単な規則または複雑な規則を定めることによりノードの間の種々の結合を行うことができる。アプリケーション302は、事象通知ターゲットと



してそれ自身を登録することによりデータベースの変化をそれが起きた時に気付くようにされることもできる。

### 【0113】

以上、本発明の特定の実施形態について説明し、示したが、種々の変更および修正を、付録Aに続く特許請求の範囲によって定められる本発明の範囲及び要旨を逸脱することなく、当業者により行うことができる。

### 【0114】

#### 付録A

PLXプロトコルは安価で、使用が容易であり、融通性に富み、信頼性があり、かつキーボードを変更でき、多数のスマート・ノードとダム・ノードを共通のデータ・チャネル／制御チャネルを通じて通信できるようにするネットワークアーキテクチャ／プロトコルである。ネットワーク化プロトコルによってこのネットワークにおけるどのようなノードもアクティブなネットワーク・サーバとして自身を割り当てることができるようにする。アクティブなネットワーク・サーバはラインアップ・カードを基にしてクライアント・ノードをポーリングする。インアクティブなノードはラインアップ・カードから自動的に除去され、したがって、不必要なポーリング・トラフィックを減少する。このアーキテクチャは衝突を減少させ、しかも実際のデータ伝送のための帯域幅を維持する。制御およびデータのネットワーク化要求のサポートがプロトコルによって行われる。ストリーミング・データまたは非同期データのためのサポートが、タイムスロットをネットワークに割り当て、かつ2つの知能ノードが、アクティブなネットワーク・サーバにより仲裁されて相互に直接通信することができるようにする。アクティブなネットワーク・サーバは、主ネットワークの動作とは独立に大量のデータ・トラフィックが流れることができるように、別々のデータ・チャネルを割り当てることもできる。アクティブなネットワーク・サーバとして動作するネットワーク・ノードはダイナミックに変更でき、かつ通常は、睡眠中のネットワークへの要求の送信を開始する第1のノードにより決定される。クライアント・ノードは、アドレス分離スキームを用いてダイナミック・ポーリングによりアドレスされる。

### 【0115】

PLXプロトコルを含んでいるPLXアーキテクチャは、建物内部の既存の電力線（電源線）をネットワーク媒体として使用するネットワークに良く適する。データの伝送に既存の電源線を使用することは、ユーザーがネットワーク・ケーブルを設置する必要がないことを意味する。

#### 【0116】

PLXアーキテクチャはネットワーク・ノードのために頑丈で、決定的な媒体アクセス可能性を提供する。ノードはアドレス分離スキームを用いるダイナミック・ポーリングによってアドレスされる。実行可能なデータ・チャネルが診断、議論のやりとり、および汎用データやり取りアプリケーションに使用するために提供される。

#### 【0117】

一実施形態では、PLXプロトコルは大局的に独自の識別コードと、ノード輪郭と、32ビット仮想アドレス可能性とを提供する。それによってPLXプロトコルはプラグーンプレイ・タイプのネットワークに適合できる。

#### 【0118】

一実施形態では、PLXアーキテクチャ・プロトコルはピアリング（peer ing）、多数サーバ、簡単な構成、安全、データグラム検出、多数のデータ・フォーマット、および優先順位決定のスキームなどの諸機能を提供する。CRCおよびチェックサムなどの誤り検出、およびデータ完全性性能はPLXのある実施形態の部分である。PLXアーキテクチャはスマート・ノードとダム・ノードを提供し、かつこのアーキテクチャは簡単な制御から複雑なデータ・ストリーミングまでの範囲のデータ・トラザクションを提供する。

#### 【0119】

一実施形態では、PLXは状態マシン論理またはマイクロ制御によって実現される。流線形の低い端部のノード（ダムノード）を、フルPLX性能のサブセットを使用するために実現できる。機器などの、中間範囲のノードはここで開示されているプロトコルに適合する。PCまたはPSX、インターコム／調査装置、プリンタ、マウス、およびその他のデータ集中ノードなどのより高い端部のノード（スマートノード）はPLXアーキテクチャにおいてまた応用性を見出す。

## 【0120】

PLXプロトコルはデータ・リンク層、ネットワーク層、およびトランスポート層のための動作規則を定める。一実施形態では、PLXはデータ・リンク層の媒体アクセス制御(MAC)部を含む。MACプロトコルは、物理的媒体を各ノードがいつ、どのようにしてアクセスできるかを支配する規則の集まりである。一実施形態では、MACプロトコルは、電源線における衝突を減少させる、集中して分布されているダイナミック・トークン・パッシング・アーキテクチャを使用する。

## 【0121】

PLXアーキテクチャによって、ネットワークにおけるどのようなノードもアクティブなネットワーク・サーバとしてそれ自身に割り当てることができるようにされている。それはトークンのための要求を仲裁する。ノードがインアクティブであると、それらのノードは「睡眠」モードに入り、したがって、不必要な任意の「ポーリング」トラフィックを無くす。このアーキテクチャは衝突を減少させ、しかも実際のデータ伝送のための貴重な帯域幅を維持する。

## 【0122】

PLXアーキテクチャは、多くの面で、クライアント/サーバネットワーク化アーキテクチャであって、制御ネットワーク化およびデータネットワーク化の需要のためのパケットをサポートする。ストリーミング・データまたは非同期データのためのサポートは、ワイヤにタイムスロットを割り当て、2つの知能ノードがアクティブなネットワーク・サーバによる仲裁で相互に直接話し合えるようにすることにより、サポートできる。アクティブなネットワーク・サーバは、大量のデータ・トラフィックを主ネットワークの動作とは独立に流れることができるように、別々のデータ・チャネルを割り当てることができる。アクティブなネットワーク・サーバとして機能しているネットワーク・ノードはダイナミックに変更でき、かつ睡眠中のネットワークで送信要求を開始する第1のノードによって通常決定される。また、アクティブなネットワーク・サーバはアプリケーション・サーバとは独立に選択される。アプリケーション・サーバは通常は固定ノード場所である。アクティブなネットワーク・サーバはノードになり得る任意のサー

バとすることができる。

### 【0123】

一実施形態では、PLXは、インアクティブな（睡眠中）ネットワーク媒体を最初にアクセスするためのデータグラム検出アルゴリズムと、その後続く、アクティブなネットワークに挿入するための集中化されたトークン・パッシングを含む、組合わされた媒体アクセス性能を提供する。これは多数のアクセスを衝突のない、トークン・パッシング・タイプの環境に実効的に結合する。これは決定論の付加された利点である。一実施形態では、PLXは最初の媒体アクセス可能性を決定するデータグラムの存在を用いる。データグラムは指定された前文／長さシーケンス組合わせを一致させることによって特に決定される。

### 【0124】

一実施形態では、PLXは、システムのアクティブなノードへトークンを通すだけである集中化されたダイナミック・ポーリング・アルゴリズムを用いることによってネットワークにおけるトラフィックを減少させる。オペランドがひとたびインアクティブになると、そのノードはポーリング・リストから除去される。この選択的なポーリング・プロセスは、それら自身を「バスの分割」として知られているプロセスによってポーリング・リストに挿入できるノードの性能を基にしている。

### 【0125】

この分割プロセスはポーリング・リストへの実時間で実行中の挿入を可能にする。この分割プロセスは複数のノード応答が単一のシステム応答として見られるようにする。このシステム応答は、アクティブなノード（ポーリングを行ってノード）が特定のノード要求挿入をポーリングリスト内にさらに分割することを可能にする。

### 【0126】

ポーリング・リストからの実時間、実行中の除去が時間が経過したプロセスにより行われる。インアクティブなノードは、所定の時点の後でそれらがトークンを使用しなければ、最終的にはポーリング・リストから除去される（挿入を解除される）。一実施形態では、時間の経過したプロセスは、ノードがトークン要求

に応答し損なったならば、更に促進される。

#### 【0127】

一実施形態では、ポーリング・リストは媒体の帯域幅性能を基にして固定サイズ（オペランドの数）に設定される。低い優先順位のデータ（照明装置の制御データなどの）を伝えるノードが、より高い優先順位のデータ（ストリーミング・オーディオ／ビデオ・データなどの）を持つノードのためにリストに余地を設けるために、ポーリング・リストから除去される。

#### 【0128】

一実施形態では、PLXアーキテクチャ内の媒体アクセス制御（MAC）層は、予備の受信バイポーラ・トランジスタ、使用中応答ハンドシェイクを用いることによって自己スロットリング機構を提供する。一実施形態では、各ノードにMACヘッダのコピーを保持するために十分大きい受け面積を設けることによって、自己スロットリングが完成される。以前の packets 要求によりノードが完全に忙殺されているとしても、その忙殺されているノードは、使用中応答を発生することにより、いぜんとして要求に応答できる。使用中応答は、伝送しているノードにその packets にバーストまたはシーケンスを近寄せなくしなければならないことを知らせて、各受信ノードの性能に従ってシステムのペースをとる。

#### 【0129】

電源投入時のノード自動アアウンス機能が遠隔のデータベース・サーバの再同期化を行う。新しいノードの電源投入時には、その新しいノードはそれが媒体に新たに到達したことをアナウンスする。

#### 【0130】

一実施形態では、PLXは好適なサーバ選択を行い、キックスタート・アルゴリズムを提供する。PLXはクライアント／サーバ型アーキテクチャであるので、媒体アクセスを仲裁するために1つのノードが通常選択される。典型的な電源線ネットワークでは、全てのノードが必ずしも等しく作られているわけではない。したがって、PLXの一実施形態によって、最も中央に（すなわち、ブレーカーパネルの近く）配置されているノードをユーザーが選択できるようにされて、好適な「アクティブなネットワーク・サーバ」として動作できるようにする。好

適なサーバがインアクティブであるとする、遠隔のノードがその好適なサーバをアクティブにすることができる。簡単な覚醒アルゴリズムによってインアクティブな好適なサーバが再び活動するようにできる。

#### 【0131】

最初に、クライアント／サーバ・モデルにおける媒体をアクセスするためにノードがトークンを獲得する。ひとたびクライアント・ノードにそのトークンが与えられると、それは媒体を指定された時間だけ使用する。この時間の期間中は、それはシステム内のどのノードとも、サーバの環境とは独立に、直接通信できる。この期間が切れると、媒体アクセス制御がサーバ・ノードに戻される。したがって、媒体仲裁がクライアント／サーバのようにしてまず行われ、それに同列間タイムスロットが続く。

#### 【0132】

一実施形態では、PLXはダイナミック媒体仲裁サーバを含む。媒体アクセスを仲裁するサーバは活動を基にしてダイナミックに割り当てられる。そのダイナミックな割り当ては、送信すべきパケットを有する最初のノードが、システムが「インアクティブである」ことを認識し、好適なサーバ（存在するならば）を覚醒させようとする数回の試みの後で、アクティブなネットワーク・サーバの役割を行う。PLXネットワークでノードになることができるどのようなサーバもアクティブなネットワーク・サーバになることができる。

#### 【0133】

一実施形態では、このネットワーク・プロトコルはストリーミング・データを電源線媒体を通じて送受信させる。一実施形態では、ストリーミング・データはデジタル音声データを含む。一実施形態では、ストリーミング・データはデジタル・ビデオ・データを含む。

#### 【0134】

一実施形態では、電源線媒体を通じて、デジタルPBX型機能および／または、デジタル・インターコム機能を提供するためにこのネットワーク・プロトコルは用いられる。このネットワーク・プロトコルは広帯域デジタルネットワーク化サービス（たとえば、DSL、ケーブル、ISDN等）を家庭内の既存の電源線

を通じて家庭全体に拡張するために用いられる。

### 【0135】

ネットワーク・プロトコルは3種類または4種類以上のネットワーク・トラフィック、すなわち、制御トラフィック、データ・トラフィック、ストリーミング・データ・トラフィック（ストリーミング・マルチメディア・データ）、を同時に扱うことができ、かつ管理できる。このネットワーク・プロトコルは所与のノード（音声装置のための決定の要求などの）のネットワーク化要求に依存して保証されたアクセス回数を許すために優先化スキームを提供する。

### PLX OSIモデル

上から5番目までのOSI層203～207の各々は大きなオーバーヘッドをネットワーク・アプリケーションに加える。図3に示すように、PLXは、共通アプリケーション言語（CAL）と呼ばれている比較的薄いアプリケーション層607と、比較的薄いトランスポート／ネットワーク層603を用いて、下側のデータ・リンク層602と物理層601を補足する。層601～603と607のおのおのはPLXコンプライアント・ノードに通常存在する。図3に示すように、PLXデータ・ネットワーク化ノード（スマート・ノード）は、アプリケーション層207と、ネットワーク層203と、トランスポート層204とに通常のOSIネットワーク性能（たとえば、TCP/IP、IPX、ウインドウズ、ネットウェア、等）を含むこともできる。PLXコンプライアント・ノードは減少させられた量の制御情報を通常含む。それは、層601～603と607において具体化されるように、PLXスタックのみを用いてPLXノードの間で送られる。

### PLX物理層

PLX物理層601はネットワーク・ハードウェアと、ネットワーク・ケーブルとに物理的にインタフェースするハードウェアの詳細を取り扱い、および、通常は、実際のハードウェア自身を含む。物理層は変調技術、使用される周波数、電力出力、等などの属性を含む。一実施形態では、PLXは下記のようにデジタル電源線（DPL）技術を用いる。

### PLXデータ・リンク層

PLXデータ・リンク層602は、アドレス指定機能、媒体仲裁スキーム、間隔の間の間隔決定、バック・オフ・アルゴリズム、などの、媒体100とのインタフェーシングの詳細を取り扱う。データ・リンク層602は、発信元アドレス／宛先アドレスと、長さと、巡回冗長検査(CRC)またはチェックサム・データなどの誤り検出／訂正データとを含むヘッダを含んでいる。

#### PLXネットワーク層

インターネット層と呼ばれることもあるネットワーク／トランスポート層603、はネットワーク上の1つの場所から別の場所へのデータの packets の経路選択を行う責任を負っている。PLX内では、ネットワーク層603はシステム、個々のノード、ソケット、およびMACヘッダ・フィールド内のネットワーク・アドレス・フィールドを用いて通常取り扱われる。

#### PLXトランスポート層

PLXネットワーク／トランスポート層603は、上に存在するアプリケーション層607のための2つのホストの間でデータの流れを行う。トランスポート層603は、一連番号および／または要求／応答型確認応答情報も含む。PLX内では、アプリケーション制御を行えるようにするために、トランスポート層603は、OSIトランスポート層203と比べて小さくされている。トランスポート層603は要求／応答接続手順アルゴリズム、再試行アルゴリズム、時間切れアルゴリズム、等を提供する。PLXはMACヘッダの制御フィールド内でネットワーク／トランスポート層603をほとんど完全に実現する。

#### PLXアプリケーション層

PLXアプリケーション層607はアプリケーションの詳細を取り扱い、パケット配信を確実に扱うために、どのトランスポートが用いられているかに依存して、アプリケーション層607は接続手順プロトコルおよび／または要求／応答プロトコルとを使用できる。フィールドのかなりの量の重複がOSI層のプロトコル内に存在する。この重複はより多くのオーバーヘッドに変わり、より多くのスペースを使用し、追加の処理パワーを要する。PLXプロトコル内では、OSIフィールドの多くが必要とされず、したがって通常省かれる。



種々のOSIプロトコル内に含まれている種々のコンポーネントを調べることにより、データ・リンク層602が上の3つの層なしに濾波の多くを行えることが明らかにされる。この濾波は、同じ通信チャネルを競合している多数のノード（たとえば、同じネットワークワイヤを競合している多数のネットワーク・カード）などの、ハードウェアの問題点に留意もするハードウェア論理にデータ・リンク層602が通常しばしば限られるため、有利である。一実施形態では、特定のネットワーク・ノードのためのネットワーク・ハードウェアが、その特定のネットワーク・ノードに宛てられたデータ・パケットを除く全てを除去する。そのようなシステムの下では、ノードはデータ・パケットのデータ部分を解析するだけである。

#### DPIのための2つプロトコル

デジタル電源線（DPL）で使用される2つのプロトコル、すなわち、低レベル・プロトコルと高レベル・プロトコル、をPLXで定めることが好ましい。低レベル・プロトコルの定義。低レベル・プロトコルはデータ・リンク層602の定義を与え、パケットが、比較的少数のネットワーク化機能およびトランスポート機能を持つ同じ媒体100からどのようにして濾波され、送られ、かつ受信されるかの定義を行う。

#### **【0137】**

高レベル・プロトコルの定義。PLXノードは少ない量の情報を含む。各PLXノードは特定のノード属性を制御する共通アプリケーション層607を使用する。これにより、ノードの種類とは無関係にPLXシステムを特徴付けることができるようにされる。アプリケーション層607はハードウェア・ヘッダが除去された後で制御情報を解釈すなわち構文解析する。

#### 物理層：デジタル電源線（DPL）仕様

PLXプロトコルは、光伝送、光ファイバ伝送、無線周波数伝送装置、ツイスト・ペア伝送装置、同軸伝送装置、衛星システム、デジタル電源線（DPL）装置、等を含めた、多くの種類のネットワーク媒体（すなわち、データ伝送装置）に使用できる多用途プロトコルである。

#### **【0138】**

電源線搬送装置としても知られているDPL装置はデジタル・データを伝送するために電源配線（たとえば、建物内の標準110ボルト交流（VAC）回路）を使用する。一実施形態では、PLXプロトコルは、単一の低速チャネル（350～1000k b p s）、約5.6MHzの低速搬送周波数、約80dBまたはそれより良いダイナミックレンジ、狭い帯域幅使用（速さに依存するが、1MHz位）のDPLに関連して使用される。

#### 【0139】

一実施形態では、PLXプロトコルは、多数の高速チャネル（合計で4～8mbps）、30MHzまたはそれより上までの高速搬送周波数、約80dBまたはそれより広いダイナミックレンジのDPLに関連して使用される。

#### 【0140】

通常のDPL装置では、送信搬送波はデータの少なくとも20マイクロ秒前にインエンプル状態にされ、受信器が搬送波を検出しなくなるまで、送信器を動作不能にする間の時間は15マイクロ秒またはそれより長くできる。

#### 低レベルプロトコル層：PLXの仕様

PLXプロトコルは、簡単な制御から複雑なデータ・ストリーミング・ネットワークまでの範囲に及ぶアプリケーションのために加減できる。一実施形態では、PLXプロトコルは一般的なCAL仕様の諸特徴のほとんどに影響を与えるようになっている。EIA-600において定められているCEBusはバス装置を制御するための産業標準制御言語である。EIA-600は家庭LAN内で使用するための共通アプリケーション言語の骨組みを提供する。一般的なCALはEIA-721規格シリーズ（EIA-721.1、EIA-721.2、EIA-721.3、およびEIA-721.4を含む）で定められている。CEBus産業協議会（CIC）は、その言語を使用するための「文法」規則を制定することによって、その骨組みに肉付けする家庭プラグ・アンド・プレイ（Home Plug & play）（HPP）仕様を制定している。

#### 【0141】

HPP仕様は、家庭内の製品およびシステムが家庭の状態を基にして動作を行えることができるようにする、それらの製品およびシステムの挙動特徴の集まり

の詳細を定めている。たとえば、その仕様は、「住人が不在」または「住人が帰宅して睡眠中」などの家庭内の種々の状態を識別して、安全システムの起動、屋内灯の消灯、または温度設定などの適切な操作を家庭システムが行えるようにする。HPP仕様は家庭制御用ウィンドウズ® 95 PCベース・アプリケーションを開発するための情報も含む。

#### 【0142】

EIA-600において定められている共通アプリケーション言語は、種々の産業部門（たとえば、娯楽、コンピュータ、冷暖房、台所設備、等）で製造された家庭LAN製品の間の通信のための枠組みを提供する。

#### 【0143】

各産業部門は、その製品がその言語を使用するための基礎として役立つ、「アプリケーション・コンテキスト」（すなわち、文法規則）を定める。CICは種々の産業部門が「調和のとれた」アプリケーション・コンテキストを開発することを支援するサポート組織として機能するために作成された。CICのHPPは、CALをベースとする相互に動作できる製品で家庭LAN市場を求めているそれらの産業部門のための、調和のとれたアプリケーション・コンテキストの要約である。

#### 【0144】

CEBus／一般的CALアプリケーションの全体を参照することによりここにその仕様が包含される。

#### 媒体アクセスの概観

PLXは、集中されたトークン・パッシング・スキーム、またはDSMA／CTPを持つデータグラム検出多重アクセス・プロトコルとして特徴付けることができる。多数の同等のノードが同じ物理的媒体100をアクセスすることを許されているために、PLXは、データを媒体100に置くことを試みた時に各ノードを使用するための共通規則群を述べる。

#### 【0145】

PLXは、種々の数のプロトコルからのいくつかの機能を統合して単一の効率的な決定論的環境を生成する。PLXはデータグラム検出を行う。各PLXノー

ドはトラフィックのための媒体100を「検出」でき、媒体100が現在優勢であるならばそれ自身を主張する。構成されたトークン・バッシング型機構を介して衝突回避が行われる。PLXは媒体に対するアクセスを取り扱うために単一の中央仲裁ノードを選択する方法を含む。その中央ノード（アクティブなサーバ）はアクティブなシステムにトークンが確実に存在するための責任を負う。PLXは、設計の簡素化と、実現の容易さと、衝突なしのアクセスと、系統的な受容と、トークンのその後の放棄と、データを確実に配信するため確認応答シーケンス（要求／応答）とを行うために選択的な動的ポーリングを行うために選択的かつ動的ポーリングを用いる。

#### 【0146】

PLXはノードが「アクティブの」時に「静止した」媒体100を持つ能力を提供する。通常は、PLXにおいては、「アクティブな」ノードのみが媒体100で通信する。PLXはプラグ-ン-プレイ性能のための全体的アドレス指定方式と、媒体100のための複数ノード競合を分離するためのアルゴリズムを提供もする。

#### 【0147】

PLXは、ストリーミング・アプリケーションのための時間決定論、または保証されたタイム・スロット、と、ターンアラウンド時間を短くするために短縮されたセル長さ（パケット長さ）とをも提供する。

#### 【0148】

PLXは多重レート・サポート・パケットと、ホット・スワッピング・パケットと、真正証明および安全パケットと、制御および管理パケットとを提供する。

#### 【0149】

また、PLXはより高い層プロトコル内に多くの制御ネットワーク化機能を提供する。その結果、媒体アクセス方法が種々のトポロジーの多くの有利な機能を利用する高度に洗練されたものになった。

#### 媒体アクセス方法論

媒体アクセス方法論は媒体100をアクセスする際に含まれる諸規則の概要を示すものである。媒体100をアクセスするPLX法は3つの事象、

1. データグラム検出または「聴取」；
2. バスの分割；
3. 集中化されたトークン・パッシング

を通常含んでいる。

#### 【0150】

ノードはシステムに存在するトークンに関して、アクティブネットワーク・サーバ・ノードとして、またはクライアント・サーバとして特徴付けられる。PLXシステムでは、媒体100に対する最初のアクセスは、活動を聴取し、その後でアクティブなネットワーク・サーバとして自己表明し、最後に、活動しているネットワーク・サーバによる系統的な、集中化されたトークン・パッシングをすることにより行われる。

#### 【0151】

図5は、どのノードが媒体100で「話す」ことを許されるかを仲裁するためにPLXにより用いられる媒体アクセス・アルゴリズムを示す流れ図である。図5に置ける流れ図は電源投入および告知処理ブロック801で始まる。そうすると各ノードは、電源を投入されると、媒体100に置けるその存在を告知される。告知が終わると、プロセスは判定ブロック802へ進む。ノードは、送信(Tx)準備完了コマンドが受け取られるまで、判定ブロック802内でループ(アイドル)し、そのコマンドが受け取られると、プロセスは判定ブロック803へ進む。この判定ブロック803において、ノードがラインアップ・カード上にないか、またはアクティブサーバでなければ、プロセスはデータグラム検出ブロック804へ進み、さもなければプロセスは判定ブロック816へ進む。判定ブロック816では、ノードがトークンを受け取っていたならば、プロセッサはパケット送りブロック814へ進み、さもなければ、プロセッサは時間切れ判定ブロック810へ進む。判定ブロック810では、時間切れが起きていなかった場合、プロセスは判定ブロック816へ戻る。さもなければ、プロセスはデータグラム検出ブロック804へ進む。パケット送りブロック814では、プロセスは送信パケットを送って、ポーリング・ブロック815へ進む。ポーリング・ブロック815では、図7を参照して説明するように、アクティブなネットワーク・

サーバがアクティブなノードをポーリングし、またはそのノードがクライアントであれば戻る。ポーリング・ブロック815が終了すると、プロセスは判定ブロック802へ進む。

#### 【0152】

データグラム検出ブロック804では、ノードは媒体100を特定の時間聴取し、その後で判定ブロック805へ進む。処理ブロック804の聴取期間中に媒体が覚醒しているならば、プロセスはLIP要求判定ブロック806へ進み、さもなければ、プロセスは処理ブロック812へ進む。処理ブロック812では、そのノードは「覚醒」パケットを送り、判定ブロック814へ進む。判定ブロック814では、3つの覚醒パケットが送られたがそれに対する応答がなかったとすると、プロセスは自己表明ブロック813へ進む。さもなければデータグラム検出ブロック804へ戻る。自己表明ブロック813では、ノードはアクティブなサーバ・ノードとして自己表明し、プロセッサはパケット送りブロック814へ進む。

#### 【0153】

LIP要求判定ブロック806では、プロセスはLIP要求の存在を調べる。LIP要求が存在しなければ、プロセスは時間切れ判定ブロック809へ進み、さもなければ、プロセスは処理ブロック807へ進む。時間切れ判定ブロック809では、プロセスは指定されたパケット時間切れ期間が経過したかどうかを調べる。その期間が経過したならば、プロセスは判定ブロック802へ進み、さもなければプロセスはLIP要求判定ブロック806へ戻る。

#### 【0154】

処理ブロック807では、ノードはバスにスピット・オンし、その後で判定ブロック808へ進む。判定ブロック808では、ノードがドラフトされたかどうかをプロセスは調べる。ノードがドラフトされたとすると、プロセスはトークン受信判定ブロック816へ戻り、さもなければ、プロセスはLIP要求判定ブロック806へ戻る。

#### 【0155】

ブロック802、803、810、および814～816は集中化されたトー

クン・パッシング・アルゴリズムの一部である。ブロック804、805、および811～813はデータグラム検出（聴取）アルゴリズムの一部である。ブロック806～809はスピッティング・オン・ザ・パス・アルゴリズムの一部である。

#### 【0156】

図5に示すように、媒体100に対する最初のアクセスは、媒体100が「睡眠中」であるか「覚醒している」かに依存して2つの異なるやり方の1つで行われる。媒体100が睡眠中であれば、アクセスを望んでいるノードはアクティブサーバとして自身を自己表明する。媒体100がアクティブ（すなわち、アクティブネットワーク・サーバにより用いられている）と、アクセスを望んでいるクライアント・ノードがアクティブなネットワーク・サーバにアクセスを求める。アクティブなネットワーク・サーバはアクセスを求めたクライアント・ノードのラインアップ・カードを保持している。クライアント・ノードは「スピッティング・オン・ザ・パス」として知られているプロセスによってラインアップ・カードに記入されることを求める。

#### 【0157】

通常は、サーバになることができるどのようなノードもアクティブなネットワーク・サーバとして自身を表明できるが、所与のノード内にサーバになることができる属性を含むという要求ではない。

#### 【0158】

アクティブなネットワーク・サーバがひとたび選択されると、それは、ポールすべきアクティブなサーバ・ノードのリストを含む「ラインアップ・カード」を作成および保持することができなければならない。アクティブなネットワーク・サーバの全てがインアクティブになり（エージング・プロセスによって）と、アクティブなネットワーク・サーバはアクティブなサーバとしてのその現在の状態を放棄し、媒体100は再び眠る（睡眠）ことになる。通常、アクティブなネットワーク・サーバは、媒体100へ送り出す何かを有するノードにより自己指名される。

#### 【0159】

アクティブノードがある時間沈黙しているとそのノードはラインアップ・カードから除去される。アクティブノードは、高い優先順位のデータを持つノードがラインアップ・カードに対するアクセスを必要とする時にもラインアップ・カードから除去される。ラインアップ・カードは通常は最大数のスロットを有する。いいかえると、ラインアップ・カードはラインアップ・カードに載せることができる最大数のノードを有する。スロットの数は媒体100で利用できる帯域幅と、種々のネットワーク・ノードにより必要とされている帯域幅とにより通常決定される。Nがラインアップ・カード中のスロットの最大数、tが特定のアクティブなノードがトークンを保持することを許される最も長い時間（ミリ秒）とすると、そのアクティブなノードは $N \times t$ ミリ秒ごとに少なくともおよそ1回トークンを獲得する。したがって、ラインアップ・カードは、アクティブなノードが定期的な、予測できるペースでポールされることになる、という決定を行う。たとえば、ストリーミング・ビデオ・データはストリーミング・オーディオよりも高い優先順位を持つ。そうすると、N個のストリーミング・ビデオ・ノードがラインアップ・カードに既に載せられていると、ラインアップ・カードに載せられることを求めているストリーミング・オーディオ・ノードは拒絶される。しかし、ストリーミング・オーディオ・ノードには、ラインアップ・カードに載せられることを求めるたびに、トークンが与えられる。ラインアップ・カードに載せられているノードは自動的にポールされ、したがって、トークンを求める必要なしにトークンを定期的に獲得する。ラインアップ・カードに載せられていないノードは、トークンに対する要求、またはラインアップ・カードに載せられる要求、を行った後でのみトークンを受け取る。

#### 【0160】

特定のネットワーク・ノードにより提供されるデータの優先順位は、以下に説明するノード・プロフィール・オブジェクトに関連して記載されたnetwork\_classフィールドにより決定される。特定のノードに対するnetwork\_classはノード・アドレスの上位4ビット(device\_typeフィールド)中にも見出される。

#### 【0161】



### ノード・セマフォ

各PLXノードは、システムの現在の状態と、システム内におけるノードの関わり合いとを反映する2つのローカル・セマフォを管理する。それらのセマフォは聴取プロセスを開始すべきか否かをノードが決定することを支援する。通常は、それら2つのセマフォは媒体100に対するアクセスを行うために用いられるので（ノードが送信するものを有しているとき）、ノードはそれらのセマフォを管理する。

#### 【0162】

最初のセマフォは「システム状態」を反映する。システム状態は、媒体100がアクティブか否か（すなわち、パケットが媒体100上で見られるか）に応じて、「覚醒している」か「睡眠中」であるかである。

#### 【0163】

第2のセマフォは「ローカル・ノード状態」と名付けられている。ローカル・ノード状態はノードの取り得る次のような3つの状態、すなわち、（1）ノードはアクティブなネットワーク・サーバ・ノードである、（2）ノードはアクティブなクライアント・ノードである、（3）ノードはノンアクティブなクライアント・ノードである、の1つを反映する。ローカル・ノード状態は、ノードが聴取アルゴリズムを開始すべきかどうか、ノードが現在ラインアップ・カードに載せられている（ポールされている）かどうか、またはノードが現在アクティブなサーバであるかどうかを判定する。

#### 【0164】

##### 「システム状態」腕木信号

各ノードは、システムが覚醒しているか、睡眠中であるかについて個々に判定する。この判定は媒体100におけるラインアップ挿入要求パケット（LIP）の存在を基にしている。ノードがLIPパケットを見ると、システム状態腕木信号は覚醒することになる。ある時間の後で、LIPパケットが見られないとすると、ノードはシステム状態を睡眠ヘトグルする。これは、アクティブネットワーク・サーバが存在するものとする、それはクライアント・ノードを覚醒状態に維持するためにLIPパケットを周期的に送らなければならないことを意味する

。

## 【0165】

ノードが媒体100を聴取せねばならないか否かを決定するために、ノードはこの腕木信号を使用する。システム状態が睡眠である時のみ、ノードは聴取プロセスを通じて媒体100を争う必要がある。

## 【0166】

「ローカル・ノード状態」腕木信号

アクティブなネットワーク・サーバは、そのラインアップ・カードに現在載せられているクライアント・ノードにトークン（ボール）を、そのクライアント・ノードによる最後の送信の後で1ないし10秒間配信する。この時点で、アクティブなネットワーク・サーバは、ノードが送信を続け、ラインアップ・カードからクライアント・ノードを「古びさせる」ことを決定する。クライアント・ノードはこれを検出できなければならない。クライアント・ノードがトークンを現在受信していると、それはアクティブとみなされる。クライアント・ノードがトークンを現在受信していないと、それはインアクティブとみなされる。インアクティブなクライアントは、アクティブなネットワーク・サーバによりラインアップ・カードに載せられた後で、「スピitting・オン・ザ・バス」と名付けられたプロセスにより、媒体100に送信することができるだけである。下の表A1には可能なノード腕木信号状態と、各状態が媒体への送信に関して意味するものを示す。

## 【0167】

システム状態	ノード状態	次の送信動作
覚醒	アクティブ	ラインアップ・カードで：トークンを待つ
覚醒	インアクティブ	ラインアップ・カード外で：スプット・オン・ザ・バス

睡眠	アクティブ	悪い状態：聴取し、その後サーバであると表明する
----	-------	-------------------------

---

睡眠	インアクティブ	聴取し、その後サーバであると表明する
----	---------	--------------------

---

表A1. 新たな送信の用意ができていないノードに対する次の動作。

【0168】

データグラム検出または「聴取」

上記システム状態腕木信号は、ノードが聴取を開始すべきか否かを決定する際の主要因である。それは、ノードがアクティブネットワーク・サーバであると自身で表明すべきか否か、またはそれがクライアントとしての服従的な役割を演ずるか否かを決定する際の主要因でもある。通常は、聴取は、睡眠中のシステムへの最初の送信に先立って行われるだけである。いずれかのノードが媒体100へ送信していると、アクティブネットワーク・サーバはLIPパケットを送るため、およびトークン配信を仲裁するために、アクティブネットワーク・サーバは既に選択されており、システムは覚醒している。システムが覚醒しているならばノードはクライアントとして行動すべきである。

【0169】

媒体100へ送る準備が整っているパケットをあるノードが有していること、およびシステム状態腕木信号が睡眠中であることをそのノードが判定すると、ノードは聴取プロセスを行ってその次のステップを決定し、この最初のプロセス中の衝突を最少にする。これは、2つのノードが媒体100を求めて争うことがあり、かつ考えられる見えていない衝突が起こり得る、PLXネットワーク上の時間のみでなければならない。このようにして、堅実なバックオフ・アルゴリズムが提供される。

【0170】

聴取においてアドレスするために次のような2つの可能なケースがある。(1) ノードに電源が投入されたばかりで、現在のシステムに加えたことを告知するために、その「告知」パケットまたは「CALーピン (ping)」パケット

を送る必要がある。(2) ノードはインアクティブで、システムを覚醒させようと試みている。いずれのケースでも、聴取中にあるサーバが検出されると、そのノードはL I Pパケットのサーチを直ちに開始すべきである。L I Pパケットはノードがラインアップ・カードをアクティブネットワーク・サーバに挿入すること、およびその後でのトークン・パッシングおよびノード送信を可能にする。

#### 【0171】

最初の「聴取／ピン (P i n g)」告知

ノードに電源が投入されると、そのノードはシステムにおけるその存在を放送C A Lーピン・パケットを送信することにより直ちに告知する。これによって、情報を「引く」ことを常に試みる代わりにそれを「押す」ことによって、自動発見機構を一層堅実なものにすることができるようになる。電源を投入されたばかりのノードはシステムに関する履歴を持っていないので、聴取アルゴリズムは正常な覚醒プロセスとは僅かに異なる。

#### 【0172】

最初の聴取はC A Lーピン・パケットを放送するまでに500msも要することがある。これは、指定された時間だけトラフィックを実際に聴取し、その後でその時間中無作為に聴取し、放送覚醒パケットを3回送信して好適なサーバに、このノードが存在するならば、それをポーリングをする機会を与えることによって行われる。この動作は3回繰り返され、それが終わると、C A Lーピン・パケットが全てのノードへ放送されて、システムへのエントリが成功したことを知らせる。聴取／ピンのための一連の動作が擬似コードで次のように与えられている。

#### 【0173】

1)

a) 125msより短い任意の長さの時間媒体100を聴取する (L I Pパケットを求める)。

b) 覚醒パケットを間隙間に300 $\mu$ sの間隔をおいて3回放送する。

c) 聴取を続けて全部で125msの時間を終了する。

2)

a) 125msより短い任意の長さの時間媒体100を聴取する(LIPパケットを求める)。

b) 覚醒パケットを間隙間に300 $\mu$ sの間隔をおいて3回放送する。

c) 聴取を続けて全部で125msの時間を終了する。

3)

a) 125msより短い任意の長さの時間媒体100を「聴取する」(LIPパケットを求める)。

b) 覚醒パケットを間隙間に300 $\mu$ sの間隔をおいて3回放送する。

c) 「聴取」を続けて全部で125msの時間を終了する。

4) アクティブネットワーク・サーバとして表明し、放送「CALーピン」パケットを送信して存在を示す。

5) アクティブネットワーク・サーバとしての表明を取消す。

#### 【0174】

上記聴取／ピン・プロセスはノードに電源が投入された後で1回行われ、そこからこのプロセスが要する待ち時間は通常は長くない。以下に説明する実行時間覚醒プロセスはもっとしばしば実行されるので、待ち時間はもっと短いことが望ましい。

#### 【0175】

実行時間「聴取／覚醒」シーケンス

ノードに電源がひとたび投入され、それがシステムに存在することが告知されると、それは実行時間モードで動作を開始する。その実行時間動作モード中に、ノードがパケットを睡眠中のシステムへ送る必要があると、それは同様な一連の事象を進んで、好適なサーバを覚醒させようと試みる。好適なサーバが存在せず、かつアクティブネットワーク・サーバが存在しないとすると、ノードはアクティブネットワーク・サーバと自身を表明し、クライアント・ノードのポーリングを開始する。聴取／覚醒アルゴリズムのための疑似コードの表が下に与えられている。下に与えられているアルゴリズムに加えて、応答時間をより短くするために、システムの状態を反映させるためにノードは媒体100の監視と、ローカル・ノードセマフォの使用とを交互に行うことができる。このプロセスに関連す

る待ち時間を一層短縮するために、ローカル・ノードセマフォは覚醒パケットに関連して用いられる。

【0176】

1)

a) 125msより通常短い任意の長さの時間媒体100を聴取する (LIPパケットを求める)。

b) 覚醒パケットを間隙間に300 $\mu$ sの間隔をおいて3回放送する。

c) 「聴取」を続けて全部で125msの時間を終了する。

2)

a) 125msより通常短い任意の長さの時間媒体100を聴取する (LIPパケットを求める)。

b) 覚醒パケットを間隙間に300 $\mu$ sの間隔をおいて3回放送する。

c) 聴取を続けて全部で125msの時間を終了する。

3)

a) 125msより短い任意の長さの時間媒体100を「聴取する」 (LIPパケットを求める)。

b) 覚醒パケットを間隙間に300 $\mu$ sの間隔をおいて3回放送する。

c) 聴取を続けて全部で125msの時間を終了する。

4) アクティブネットワーク・サーバとして表明し、それに応じて次のパケットを送信する。

5) アクティブネットワーク・サーバとしての表明を取消す。

【0177】

スピットィング・オン・ザ・バス

システム上のノードが送信準備の整ったパケットを有し、かつシステムが覚醒している (アクティブネットワーク・サーバが存在し、かつそれが現在トークンを配信している) 時に「スピットィング」プロセスは起きる。アクティブネットワーク・サーバは媒体100をアクセスすることを許可されている唯一のノードである。アクティブネットワーク・サーバのラインアップ・カードは、インアクティブのクライアント・ノードが媒体100をアクセスできるようにする機構の

ことである。

#### 【0178】

典型的な実行時間動作中は、ネットワークは2つの状態、すなわち、睡眠中または覚醒中、のいずれか1つで現れる。スピッティング・プロセスはネットワークが現在どの状態にあるかに応じて僅かに異なる。

#### 【0179】

##### 睡眠および覚醒状態

サービスを現在求めているノードがないと、アクティブネットワーク・サーバが判定した時にネットワークは睡眠状態に入り、その結果、トークンの送信を停止する。ネットワークの動作停止の前に、アクティブネットワーク・サーバは一連のマスクされた群LIP (LIPG) 要求パケットを指定された時間送る。一連のLIPG要求パケットがどのクライアント・ノードからも応答を引き出さないと、アクティブネットワーク・サーバは活動しなくなり、ネットワークは睡眠状態に入る。送信を求めているノードによるネットワークへのその後のエントリは、その後で上記正常な競合取扱い、聴取アルゴリズムを介して行われる。

#### 【0180】

覚醒状態は、指定されたネットワークにおける、1つまたは複数の遠隔ノードと情報を能動的に交換しているノードを象徴する。覚醒状態では、媒体アクセスはアクティブネットワーク・サーバおよびそのラインアップ・カードにより制御される。ラインアップ・カードに現在載せられているノードに対するトークン・パッシングを用い、かつラインアップ・カードに載せられようと試みているノードに対するスピッティングによって衝突は減少させられる。

#### 【0181】

##### 「スピッティング・オン・ザ・バス」シーケンス

スピッティング・オン・ザ・バスのためのシーケンスによって、アクティブネットワーク・サーバがLIPGパケットを周期的に送ることができるようにされる。睡眠中のクライアント・ノードはLIPGパケットに応答することを許される。応答がひとたび見られると、アクティブネットワーク・サーバはマスクされていないLIPD要求を全てのノードへ送って、トークンを望んでいるノードの

アドレスを持つ単一の応答を希望する。2つ以上のノードがトークンを求めて競合しているとする、応答は見られず、アクティブネットワーク・サーバはノード分離シーケンスに入る。

#### 【0182】

図6Aおよび図6Bは、それぞれアクティブネットワーク・サーバおよびクライアント・ノードのためのスピitting・オン・ザ・バスのプロセスを示す図6Aでは、アクティブネットワーク・サーバのためのスピitting・オン・ザ・バスのプロセスは、ノードがアクティブネットワーク・サーバになった時にスタート・ブロック901で始まる。プロセスはスタート・ブロック901からポーリング・ブロック902へ進む。ポーリング・ブロック902では、アクティブサーバは現在ラインアップ・カードに載せられているクライアント・ノードの全てをポーリングする。ポーリングがひとたび終了すると、プロセスは送信ブロック903へ進む。送信ブロック903では、アクティブサーバ・ノードはマスクされていないLIPG要求を送り、その後で判定ブロック904へ進む。判定ブロック904では、アクティブサーバはLOGI応答を探す。LOGI応答が受けられたとすると、プロセスはプロセス・ブロック905へ進み、さもないければ、プロセスはポーリング・ブロック902に戻る。

#### 【0183】

プロセス・ブロック905では、アクティブサーバはマスクされていないLIPD要求を送り、その後で判定ブロック906へ進む。判定ブロック906では、アクティブサーバが直接ACK(DACK)応答を探す。1つのDACK応答が受信されると、プロセスはプロセス・ブロック907へ進む。多数のDACK応答が受信されるか、DACK応答が受信されないと、プロセスはノード分離ブロック910へ進む。プロセス・ブロック907では、DACK応答を送ったクライアント・ノードがラインアップ・カードに加えられ、その後でプロセスはポーリング・ブロック902に戻る。

#### 【0184】

プロセス・ブロック910(ノード分離アルゴリズムを開始している)では、プロセスはLIPGマスクを初期化し、プロセス・ブロック911へ進む。プロ



セス・ブロック911では、マスクが更新され（たとえば、マスク中の次のビットがトグルされる）、プロセスは送信ブロック912へ進む。送信ブロック912では、マスクされたLIPG要求が送られ、プロセスは判定ブロック913へ進む。判定ブロック913では、プロセスはLOGI応答を探す。LOGI応答が受信されると、プロセスは判定ブロック915へ進み、さもなければ、プロセスはプロセス・ブロック914へ進む。プロセス・ブロック914では、プロセス・ブロック911で最近にトグルされたマスク・ビットがトグルを解除され、プロセスは判定ブロック915へ進む。

#### 【0185】

判定ブロック915では、マスク中の全てのビットがトグルされると、プロセスはプロセス・ブロック916へ進み、さもなければ、プロセスはプロセス・ブロック911に戻る。プロセス・ブロック916では、アクティブネットワーク・サーバはマスクされたLIPG要求を送り、判定ブロック917へ進む。判定ブロック917では、DACK応答が受信されると、プロセスはプロセス・ブロック907へ進み、さもなければ、プロセスはポーリング・ブロック902に戻る。

#### 【0186】

プロセス・ブロック903～907は最初のシーケンスをスピittingして  
いるサーバの一部である。プロセス・ブロック910～917はノード分離シー  
ケンスをスピittingしているサーバの一部である。

#### 【0187】

図6Bはクライアント・スピitting・アルゴリズムを示す流れ図であって、  
アクティブネットワーク上のクライアントのためのスタート・ブロック931  
で始まる。スタート・ブロック931から、プロセスは判定ブロック982へ進  
み、そこで送信状態が調べられる。送信状態が「準備完了」であれば、プロセス  
は判定ブロック983へ進み、さもなければプロセスはアイドル・ブロック98  
8へ進む（アイドル・ブロックは判定ブロック982に戻る）。

#### 【0188】

判定ブロック983では、ノードがシステム・トークンを受信すると、プロセ

スは送信ブロック989へ進み、さもなければ、プロセスは判定ブロック984へ進む。送信ブロック989では、ノードはデータの packets を送り、プロセスは判定ブロック982に戻る。判定ブロック984では、ノードがLIPD要求を受信すると、プロセスはプロセス・ブロック990へ進み、さもなければプロセスは判定ブロック986へ進む。判定ブロック986では、プロセスは時間切れまたはシステム睡眠状態を調べる。プロセスが時間切れまたは睡眠を検出すると、プロセスはプロセス・ブロック987へ進み、そこで現在のノードが実施例をアクティブサーバであるとして表明する。

#### 【0189】

プロセス・ブロック990では、LIPDからのマスクが現在のノードのノード・アドレスと比較され、プロセスは判定ブロック991へ進む。判定ブロック991では、マスクがノードに一致すると、プロセスは応答ブロック992へ進み、さもなければ、プロセスは判定ブロック982に戻る。応答ブロック992では、ノードはネットワーク・サーバに応答し（適切なLOGまたはDACKで）、プロセスは判定ブロック682に戻る。

#### 【0190】

##### 群LIP (LIPG) 質問

ネットワークが覚醒している間、アクティブネットワーク・サーバは群LIP質問を周期的に放送する。群LIP (LIPG) 質問は論理群分離 (LOGI) 応答を任意の数のノードから求める。この機構は、衝突のない機構において使用中ネットワーク内にラインアップ・カードに挿入する機会をクライアント・ノードに与える。LOGIパケットの利点は、多数の同時ノードがこの種のパケットを送信できること（それらが同じ時間内にあると仮定して）、および結果が単一のLOGIパケットである、ということである。したがって、多数のLOGI応答によって単一のLOGIパケットが受信ノードにより見られることになる。

#### 【0191】

最初のLIPシーケンス・パケットは、ネットワークにおけるいずれか1つのノードがLIPシーケンスをラインアップ・カードに挿入することを開始することを望んでいるかどうかを判定するために送られる、マスクされていない群LI

P (L I P G) 質問である。L o G I シーケンスが見られたとすると、おそらく単一のノードのみが挿入することを望み、したがって、マスクされていない直接 L I P (L I P G) パケットが次に送られる。直接応答が見られないと、その後の L I P G パケットがマスクされたアドレスを持つ群パケットとして送られる。これは、ラインアップ・カードに挿入される特定のノードを分離するために用いられる、骨の折れる、効率の低い分離機構である。これはビットマスクを系統的に送ることによって行われる。それは遠隔ノードの32ビットアドレスの単一のビットを同時に分離する。2つまたは3つ以上の衝突したノードがトークンを同時に要求したときに、この分離機構は実行されなければならない。

#### 【0192】

##### 直接 L I P (L I P D) 質問

直接 L I P (L I P D) 質問は L I P G 質問の結果として送られる。L I P D 質問の目的は、単一のノードのみが応答することを希望して（それは時間のほとんどのケースでなければならない）マスクされていない L I P D 要求を全てのノードへ送ることにより L I P プロセスを促進することである。L I P D パケットは応答するノードのアドレスを含む通常の D A C K 応答で応答される。単一のノードが応答すると、応答が見られ、ノード・アドレスがラインアップ・カードに適切に加えられる。しかし、L I P D 要求が見られないと、（多数のノードが同時に応答するために）アクティブネットワーク・サーバは、L I P G パケットを用いて、正常な分離アルゴリズムによって分離を続け、「ラインアップ・カード」に挿入される競合しているノードのうちのただ1つを選択する。

#### 【0193】

したがって、単一のノードだけが要求に応答することを望んで、L I P D パケットは分離プロセスを促進するために用いられるだけである。

#### 【0194】

##### ノード分離シーケンス

ノードが最初の L I P G に応答したが、なんらかの理由で単一の応答が L I P D 質問からは見られないと、アクティブネットワーク・サーバはノード分離に自動的に進む。分離シーケンスは、L o G I 応答を要求する L I P G パケットを使

用する。これによって多数の同時応答をアクティブネットワーク・サーバによって見られるようになる。

#### 【0195】

「アクティブネットワーク・サーバ」は、第1のアドレス（最下位）ビットがセットされたパケットを送ることによってこのシーケンスを開始する。送信を望んでいるノードは、この特定のアドレス・ビットがそれら自身に一致し、かつ一致した時のみ、このパケットに応答する。このアルゴリズムは、元のマスクとの比較が後に続く簡単な「AND」である。2つの値が一致すると、パケットはL o G Iにより応答される。

#### 【0196】

アクティブネットワーク・サーバはその後以前に一致したマスクをそのまま持ち、次のビットがセットされている次のパケットを送る。再び、ビット・シーケンスが一致したならばノードは応答する。どのノードも応答しないと、アクティブネットワーク・サーバは現在のビットをクリヤし、そのパケットを再び試行する。これは、32ビットの全てが識別されて一致が見出されるまで続行される。この時点で、唯一に識別されたノードがアクティブネットワーク・サーバのラインアップ・カードに加えられる。

#### 【0197】

集中化されたトークン・パッシング（ポーリング）

システムが覚醒していると、ラインアップ・カードに含まれている各ノードに（スピッティング・プロセスにより）、媒体100をアクセスできる時間である決定論的タイムスロットを与えることが望ましい。各ノードに使用中の媒体100を介して送信する同じ機会を与えることも望ましい。イーサネットは上記利点のいずれも欠いているが、トークン・リングは両方とも有する。

#### 【0198】

トークン・リングは、各ノードにその上流側および下流側の近くのノードのアドレス、およびトークンが常に存在／回転することを要求する点が欠点である。従来のトークン・リング・ネットワークのオーバーヘッド要件は、PLXにより意図されているダム・ノードとは適合しない。さらに、電源線ネッ

トワークの特別なネットワーク化要件はそのような厳密なトークン回転には寄与しない。したがって、PLXはダイナミック・ラインアップ・カードを有する集中化されたトークン・パッシング（CTP）機構を導入する。

#### 【0199】

CTPでは、アクティブネットワーク・サーバ・ノードは、トークンが存在すること、トークンを必要としている全てのノードがそれを得ること、睡眠中のノードが覚醒できかつトークンを受信することができること、およびトークンが決定論的なやり方で公平に分配されることを確実にを行う責任を負っている。CTPの下では、アクティブサーバ以外のノードはクライアントと呼ばれている。アクティブネットワーク・サーバの役割は上記データグラム検出または聴取プロセスを通じて自己指名される。アクティブネットワーク・サーバの役割は媒体100での所定の時間の動作の後で破棄される。一実施形態では、アクティブサーバの役割は約5秒の動作しなかった後で放棄される。システムの動作中は、アクティブネットワーク・サーバはラインアップ・カード中の各クライアント・ノードをポーリングするとともに、新しいノードに、それら自身をラインアップ・カードにスピitting・プロセスを通じて挿入できる機会を与える役目をする。

#### 【0200】

図7はネットワーク・サーバ・ポーリング・アルゴリズムを示す流れ図であって、ノードがアクティブサーバになるスタート・ブロック1001で始まる。プロセスはスタート・ブロック1001から判定ブロック1002へ進み、その判定ブロックで周期的LIPパケットを送信の必要性を判定する。LIPパケットが必要とされると、プロセスはプロセス・ブロック1010へ進み、さもなければ、プロセスはプロセス・ブロック1003へ進む。プロセス・ブロック1010では、ノードは図6Aに関連して説明したアクティブサーバ・スピitting・プロセスを実行する。プロセス・ブロック1010が終了すると、プロセスはプロセス・ブロック1003へ進む。

#### 【0201】

プロセス・ブロック1003では、プロセスはラインアップ・カード中の次のエントリを獲得して判定ブロック1004へ進む。プロセス・ブロック1004

では、ラインアップ・カード中の全てのエントリが処理されると（すなわち、全てのクライアント・ノードが話す機会を持っていると）、プロセスはプロセス・ブロック1011へ進み、さもなければ、プロセスはプロセス・ブロック1005へ進む。プロセス・ブロック1011では、トークンはアクティブサーバに与えられ（したがって、アクティブサーバが話すことを許される）、プロセスはプロセス・ブロック1005へ進む。

#### 【0202】

プロセス・ブロック1005では、トークンがラインアップ・カードから得られた次のノードに与えられ、プロセスは判定ブロック1007へ進む。判定ブロック1007では、応答時間切れが起きると、プロセスはプロセス・ブロック1012へ進み、さもなければ、プロセスは判定ブロック1007へ進む。判定ブロック1007では、クライアント・ノードがそのトークンを使用しなかった場合、プロセスはプロセス・ブロック1012へ進む。プロセス・ブロック1012では、アクティブノードの数のカウントがデクリメントされ、プロセスは判定ブロック708へ進む。

#### 【0203】

判定ブロック1008では、全てのノードがインアクティブであると、プロセスはプロセス・ブロック1009へ進み、さもなければ、プロセスは判定ブロック1002に戻る。プロセス・ブロック1009では、アクティブサーバは活動してい無いクライアント・ノードへ戻る。

#### 【0204】

##### パケットの型およびフォーマット

PLXネットワークにおけるパケットはパケットの目的に応じて種々のフォーマットをとることができる。種々のフォーマットは3つの別々の部類にグループ化するのが好都合である。

#### 【0205】

1つのフォーマットによって、多数のノードが、干渉問題や復調問題なしに、同じ応答パケットを同時に送信／受信できる。それらは論理的グループ分離（LOGI）パケットと呼ばれており、放送／再放送および確認応答のために主とし

て用いられる。

#### 【0206】

生データ・ペイロード・パケットおよびコマンド・ペイロード・パケットと呼ばれている、他の2つの型のパケットは、単一のノードが任意の所与の時点でワイヤにより通信する時に用いられる。新しい生データ・ペイロード・パケットは、そのアプリケーションに属する情報の送信／受信を望んでいるアプリケーションにより用いられる。ホスト・ノードから来るパケットは生データ・ペイロード・パケットおよび任意のCALパケットである。

#### 【0207】

PLXコマンド・ペイロード・パケットが媒体のアクセスおよび流れを管理するために用いられる。PLXコマンド・パケットはアダプタのファームウェア内およびハードウェア内で発生し、かつ終息して、ホスト・ノードは通らない。PLXコマンド・パケットはトークン、確認応答、ラインアップ挿入等の円滑な流れを容易にするものであって、全てのPLXネットワークに固有のものである。

#### 【0208】

##### 論理グループ分離 (LOGI) 応答パケット

ノードがグループ要求 (多数の同時応答の可能性を持つ要求) をネットワークへ送り出す時に第1の形式が用いられる。PLXは衝突が減少した、またはある場合には衝突がない環境が望ましいので、衝突を検出することは困難である。したがって、同時応答が可能である。図8に示されている、LOGIパケット1100は2バイト0フィールドと、その後に続く多数の2バイトオール「1」フィールドと、その後の2バイト0フィールドとを含む。この種のパケット内に存在するデータは非常に秘密であるが、群応答を単一のノードに分離することを助けるその目的を果たす。

#### 【0209】

マスクされたLIPG要求がLOGIパケットに先行する。2つ以上のノードのマスク手段はマスクされたアドレスに一致でき、したがって、多数の同時応答が生ずることができる。LIPGパケットについては後で説明する。

#### 【0210】

L o G I パケットは、特定のパケット中に存在する1の列を長くすることによってある非常に単純化されたデータも含む。長くされたパケットは、異なる型の応答を示すために、時間変位 (time displacement) と共に用いなければならない。放送パケットはこの特徴を用いて、使用中の応答を1つまたは複数のノードにより同時に示すことができるようにする。

#### 【0211】

##### ペイロード・パケット

第2の形式はネットワークの周囲でペイロードを運ぶために用いられる。これはネットワーク上で最も一般的に用いられている形式であって、有用なデータ情報を送信および受信するために有効な形式である。

#### 【0212】

ペイロード・パケットは、受信側の範囲と、それらが受信することを期待している応答の型とを示す2つの形式もとる。それらはグループアドレスされる（通常は放送パケット）パケット型および直接にアドレスされるパケット型である。群アドレスされるパケットはL o G I 応答を受信することのみができ、一方、直接アドレスされるパケットは、単一の応答が期待されているために、直接確認応答パケットすなわちDACKパケットを受信する。

#### 【0213】

ペイロード・パケットは、パケット内でのペイロードの使用を決定する2つの別々の部類にさらに分割される。それらは生データ・パケットとPLXコマンド・パケットである。

#### 【0214】

##### 生データ・パケット

生データ・パケット1200のフォーマットが図9に示されている。それはプリアンブル・フィールド1201と、長さフィールド1202と、長さフィールド1203と、ctrlフィールド1204と、宛先アドレス・フィールド1205と、発信元アドレス・フィールド1206と、シーケンス・フィールド1207と、認証フィールド1208と、DSkフィールド1209と、SSkフィールド1210と、ペイロード・フィールド1211と、CRCフィールド12



12とを含む。生データ・パケット1200はアクティブサーバ・ノードまたはクライアント・ノードによって送られる。長さフィールド1202と、長さフィールド1203と、ctrlフィールド1204と、宛先アドレス・フィールド1205と、発信元アドレス・フィールド1206と、シーケンス・フィールド1207と、それから認証フィールド1208と、DSkフィールド1209と、SSkフィールド1210はMACヘッダ1215の構成要素である。ペイロード・フィールド1211は適切なペイロード・ハンドラにより調べられるべきアプリケーション層情報を含む。ホストPCおよびCALインタープリタはペイロード・ハンドラの例である。一実施形態では、生データ・パケット1200は3バイトのプリアンブル1201と、13~15バイトのMACハンドラ1215と、255バイトまでのペイロード部1211と、2バイトのCRC1212とを有する。

#### 【0215】

PLX (外部) コマンド・パケット

PLXコマンド・パケットは、2つのノードが短いパケット・シーケンスによって通信するための手段を提供することにより、媒体100を介するデータの流れを容易にするために用いられる。PLXコマンド・パケットの変形についての説明を以下に行う。

#### 【0216】

トークン・パケット: PLXトークン・パケット1300のフォーマットが図10に示されており、それはプリアンブル・フィールド1201と、長さフィールド1202と、長さフィールド1203と、ctrlフィールド1204と、宛先アドレス・フィールド1205と、発信元アドレス・フィールド1206と、CRCフィールド1212とを含む。長さフィールド1202と、長さフィールド1203と、ctrlフィールド1204は(16進)値0,05、0,05、および0,17をそれぞれ有する。

#### 【0217】

トークン・パケット1300は直接にアドレスされたノードへ送られ、いずれのペイロード型パケットも求める。注意を必要としないノードは単にDACKす

べきであり（状態フィールドが0x03にセットされている）、これは、それらのノードが話すべきなものも持たず、トークンを使用しないことを意味する。

#### 【0218】

クライアント・ノードはアクティブネットワークへ送信する前にトークン呼び出すべきである（LIPプロセスによって）。ノードがこのトークンを用い続けている限り、アクティブネットワーク・サーバはそれにトークンを送り続ける。しかし、クライアント・ノードが「使用されていないトークン」応答で繰り返し応答すると、アクティブネットワーク・サーバはノードを老化させ、それはラインアップから外される。

#### 【0219】

トークン・パケットは通常のMACヘッダ（発信元アドレスを差し引いている）とCRCを含んでいるが、データ・フィールドは用いられない（データ・フィールドのサイズは零である）。トークンは、アドレスを0xffffffeに固定すべきである「アクティブネットワーク・サーバ」から来ることができるのみである。

#### 【0220】

直接確認応答（DACK）パケット：PLXトークン・パケット1400のフォーマットが図11に示されており、それはプリアンブル・フィールド1201と、長さフィールド1202と、長さフィールド1203と、ctrlフィールド1204と、宛先アドレス・フィールド1205と、状態フィールド1401と、CRCフィールド1212とを含む。長さフィールド1202と、長さフィールド1203と、ctrlフィールド1204は（16進）値0x06、0x06、および0x07をそれぞれ有する。

#### 【0221】

DACKパケットはパケットまたはパケット列の有効な受信に対して確認応答するために受信ノードにより送られる。DACKパケットは直接にアドレスされたメッセージ・パケット（LIPDパケットを除き）から戻されるだけである。

#### 【0222】

DACKパケットはネットワークにおける2つのノードの間の典型的なハンド

シェイキング・シーケンスを終了させるために用いられ、その結果3つのノード  
 . . . 1) アクティブネットワーク・サーバ、2) 要求するノード、3) 応答するノード、を含む。(要求するノード/応答するノードは、「アクティブネットワーク・サーバ」が現在の要求の宛先であるならば、「アクティブネットワーク・サーバ」とすることもできる)。DACK状態フィールドはパケットを受信するノードの型(アクティブネットワーク・サーバまたはクライアント)に応じて変化する。要求するノードへ(応答するノードによって)送り返されたDACKパケットは、要求しているノードへ制御を戻してパケットの流れを継続し、(要求するノードにより)「アクティブネットワーク・サーバ」へ送り返されたDACKパケットは「アクティブネットワーク・サーバ」へ制御を戻して、パケットの流れの終りを知らせる。要求するノードは、シーケンスまたはDACKパケットが受信されていないと、パケットを再度要求する役目を有する。

### 【0223】

DACKパケットは通常のMACハンドラおよびCRCと、1バイト・ペイロードとを含む。状態フィールドは受信したパケットに関する情報を含み、それはこのフィールド内に戻される。状態フィールド1101の値を表A2に示す。

### 【0224】

#### DACK   ノード   摘要

0x0	全部	バッファ・フル受信(障害)
0x1	全部	障害(多チャネル応答)
0x2	サーバ	ノードによって用いられるトークン
0x3	サーバ	ノードによって使用されないトークン
0x4	サーバ	「覚醒している」要求に応答するトークン
0x9	全部	プリンタ・シーケンス番号付けエラー
0xa	全部	プリンタ・プラグを抜かれたエラー
0xb	全部	プリンタ・オフラインエラー
0xc	全部	プリンタ一般的なエラー
0xd	全部	紙エラーのプリンタ
0xe	全部	プリンタの未知のエラー

0 x f 全部 成功

表A2. DACK状態フィールド1101の値

【0225】

この情報は実際の媒体100自体へ送られること、およびホスト・ノードにまで送られる状態ではないかもしれないことに注目すべきである。ホストに送られる状態情報に関するより多くの情報については内部PLXパケット、Tx状態についての節を参照されたい。

【0226】

ラインアップ挿入パケット (LIPDおよびLIPG) : 図12はPLXパケット1500のフォーマットを示すものであって、それはプリアンブル・フィールド1201と、長さフィールド1202と、長さフィールド1203と、ctrlフィールド1204と、宛先アドレス・フィールド1205と、マスク・フィールド1501と、CRCフィールド1212とを含む。長さフィールド1202と、長さフィールド1203と、ctrlフィールド1204は(16進)値0x09、0x09、および0x23をそれぞれ有する。

【0227】

図13はPLX LIPDパケット1600のフォーマットを示すものであって、それはプリアンブル・フィールド1201と、長さフィールド1202と、長さフィールド1203と、ctrlフィールド1204と、宛先アドレス・フィールド1205と、零フィールド1601と、CRCフィールド1212とを含む。長さフィールド1202と、長さフィールド1203と、ctrlフィールド1204は(16進)値0x09、0x09、および0x23をそれぞれ有する。

【0228】

ラインアップ挿入パケット (LIP) は「アクティブネットワーク・サーバ」によって送られて、新規の物が既存のラインアップ・カードに入ることができるようにする。これは2つの別々のパケットで行われ、それらのパケットは両方とも全ての聴取しているノードへ放送される。第1のパケット、すなわちLIPGパケット1500、はLIPマスク・フィールド1501を含む。マスク150

1はLOGIシーケンスで応答する前に遠方のもののアドレスに一致しなければならない。第2の packets、すなわちLIPD packets 1601は、応答するノードを、その発信元アドレス（ラインアップ・カードに挿入すべき）を含んでいるDACK packetsで応答させることによって、挿入プロセスを促進するために用いられる。

### 【0229】

したがって、LIPG packetsはマスクされ、かつLIPマスク・フィールド内に対応するビット・シーケンスを有する。ノードはLOGI packetsでLIPG packetsに応答しなければならない。同様に、LIPD packetsはマスクされておらず、これは、ラインアップ・カードに入ることを望んでいる全てのノード（これはノードがラインアップ・カードに既に無いことを意味している）もDACKで応答すべきであることを意味する。

### 【0230】

ペイロード・ packets・フレーム・フォーマット

下記は、ペイロード型 packets内に存在できる各フィールドについての説明である。これは、生データ・ packets型とPLXコマンド・ packets型の両方についてあてはまる。

### 【0231】

プリアンブル／スタート・シーケンスは packets・フォーマットの一部ではないが、ハードウェアを入来 packetsに同期させるキャリヤを検出するため、および現在の packets内の以後のバイトのビット時間すなわちライン速さを決定するために用いられる所定のビット・パターンである。プリアンブルの長さは、ライン上に有効なキャリヤおよび同期化を存在させるために必要な最小量のビット時間によって決定される。そのプリアンブル1201のビット・パターンは

値	シーケンス
0 x a a	第1の同期バイト
0 x 3 1	第2の同期バイト
0 x n n	速さ／第3の同期バイト

である。

## 【0232】

速さ（すなわち第3の同期）バイトは入来するデータ（長さバイト1202でスタートする）の速さを決定するものであって、次のように要約される。

## 【0233】

値	速さ
0x55	低速-350k
0xdd	中速-700k
0x99	高速-1.19m
0x11	予備

最後に、プリアンプルの後に、パケットの長さを記述する2つの重複長さバイト1202～1203が続く。それらのバイトは新たな速さで来る。

## 【0234】

## 長さフィールド

長さフィールド1202～1203は入来パケットのサイズを示すために用いられる。長さフィールド1202～1203は、有効なパケット受信を決定するためにハードウェアにより使用される（キャリヤ検出信号が存在しない中で）。パケットの長さにひとたび達すると、CRCフィールド1212の妥当性が検査される。したがって、PLXパケットの長さは全部で256バイト（プリアンプル・フィールド1201とCRCフィールド1212を含めて）に制限することが好ましい。その長さはMACヘッダ1215（制御、アドレス等）と、オプションのフィールドと、ペイロード・フィールド1211とを含む。

## 【0235】

入来データストリーム（それはプリアンプルの延長として機能する）の有当性を確実にするために長さフィールドは2つの時間（1202、1203）に重複される。長さフィールド1202～1203はパケット受信の開始前に相互に（およびプリアンプル一致）一致しなければならない。

## 【0236】

## 制御フィールド

上で示したように、ペイロード・パケットは次の2つの主な型：PLXコマン

ド・パケットまたは生データ・パケット、の1つとすることができる。

### 【0237】

PLXコマンド・パケットの型は2つの下位型、すなわち外部PLX・コマンドおよび内部PLXコマンドにさらに分類される。内部PLXコマンド・パケットは、ローカル接続（USB、1234、直列等）を介するハードウェアとホスト・ノード・ドライバとの間のハンドシェイクを指す。外部PLX・コマンド・パケットは、媒体100アクセスを調整する電源線媒体100自体上のハンドシェイク・パケットを指す。

### 【0238】

制御フィールド1204は図示のようにパケットの型に応じて変化し、各ビットは表A3に示されているように特定の定義のための専用である。

### 【0239】

ビット	<u>PLX (EXT)</u>	<u>PLX (INT)</u>	<u>RAW (NON-PLX)</u>
0:	PACKET__TYPE(1)	PACKET__TYPE(1)	PACKET__TYPE(0)
1:	PLX__SUBTYPE(1)	PLX__SUBTYPE(0)	RAW__ACK__TYPE0
2:	PLX__ACK__TYPE	予備 (0)	RAM ACK TYPE1
3:	予備 (0)	予備 (0)	暗号
4:	EXT__SUBTYPE	INT__SUBTYPE	ソケット
5:	EXT__SUBTYPE	INT__SUBTYPE	予備 (0)
6:	EXT__SUBTYPE	INT__SUBTYPE	PID
7:	EXT__SUBTYPE	INT__SUBTYPE	予備 (0)

表A3. 制御フィールド1204内のビット。

### 【0240】

#### パケット型

パケット型ビットは、所与のパケットがPLXの型か生データの型か、または非PLXの型かを示すために用いられる。PLXプロトコル要求は異なっており取り扱われ、かつほとんどの場合にマイクロコントローラファームウェアによって取り扱われ、生データ・パケットは別々のアプリケーション・ソフトウェアすなわちホスト・ソフトウェアにより通常取り扱われるので、制御フィールド内で区別

を行うと都合が良かった。生データ・パケットは、適切なアプリケーション・ソフトウェアに渡すべき生ペイロード情報を通常含んでいる。この場合の例外は、マイクロコントローラ内のインタープリタの一部と、ホスト・マシン内の一部を含んでいるCALパケットである。

#### 【0241】

<u>ビット</u>	<u>パケット型</u>
1	PLXコマンド・パケット=1
0	生データ・パケット=0

#### PLX副パケット型

PLXコマンドは2つの形式の1つで通常来る。第1の形式は他のノードによるワイヤからの要求であり、第2の形式はワイヤへは進まないホストからの要求である。マイクロコントローラファームウェアはそれら2つの型に対する応答を弁別し、かつ2つの型は相互に完全に別個なので、このビットが作成されたものである。

#### 【0242】

<u>ビット1</u>	<u>PLX副パケット型</u>
1	外部PLXコマンド・パケット=1
0	内部PLXコマンド・パケット=0

#### PLX ACK型

トークン・コマンド・パケットおよびDACKコマンド・パケットはアクセス権を媒体100へ転送するために用いられ、「アクティブネットワーク・サーバ」が媒体100の制御を他のノードへ一時的に放す場合にシーケンスを終了する。他の2つのPLXコマンド・パケット、LIPGとLIPD、は応答パケットを求める。応答型は型LOGIまたは型DACKのいずれかである。このビットはどの型の応答をノードが利用すべきであるかを決定する。

#### 【0243】

<u>ビット2</u>	<u>PLX ACK型</u>
1	DACK=1での応答
0	LOGI=0での応答



## PLX副パケット外部型

PLXの仕様は、集中化された（サーバが仲裁されたトークン）トークン・パッシング・システム内の2つのノードの間の接続なしの、確認応答されたデータ転送指すおよび確認応答されなかったデータ転送サービスを提供する。それらのビットによってこの通信が行われることが可能になる。

## 【0244】

アクティブネットワーク・サーバは、クライアントが送信を開始できる前に指令されたトークンを媒体100に置く。クライアント・ノードが媒体100に対するアクセス権を、DACK応答パケットがアクティブネットワーク・サーバ・ノードへ戻ることを指令された状態で終了する。アクティブネットワーク・サーバは、クライアント・ノードをポーリングする時にアクティブノードのラインアップ・カードを維持する。ラインアップ・カードに載せるために、クライアント・ノードは指令されたLIP要求（LIPD）または群LIP要求（LIPG）のいずれかに適切に応答する。

## 【0245】

ラインアップ・カードにおいて一旦、ノードがポールされ、かつペイロード情報を、確認応答された形式または確認応答されない形式で持っているパケットをそれらのノードは送受信できる。下記は媒体100において許された有効なPLX副パケット外部型の表である。

## 【0246】

<u>ビット（7、6、5、4）</u>	<u>バイト値</u>	<u>パケット副型</u>
0 0 0 0	0 x 0 7	DACK
0 0 0 1	0 x 1 7	トークン
0 0 1 0	0 x 2 7	LIPD
	0 x 2 3	LIPG

その他

予備

注：DACK/GACKが所定の間隙間隔要件内の要求しているノードにより受信されない場合、送信（要求または応答している）ノードがその要求（応答）を再試行する責を負う。

## 【0247】

## PLX副パケット内部型

PLX仕様によってプロトコルの一部分が、PCなどのホスト・ノードに存在できる。ホスト・ノードは、それが物理的に接続されている、取り付けられたノードに関する情報をアクセスすることを周期的に必要とする。これは、取り付けられているノード上のためであり、かつ遠方のノードへ送られるように通常はワイヤに置くべきでないため、内部PLX要求として知られている。下記は可能な内部PLX副型についての記述である。

## 【0248】

<u>ビット (7、6、5、4)</u>				<u>バイト値</u>	<u>パケット副型</u>
1	1	1	1	0 x f 1	エラーハンドシェイク
0	0	0	1	0 x 1 1	CAL要求
0	0	1	0	0 x 2 1	CAL応答
0	0	1	1	0 x 3 1	Tx状態
1	1	x	x		予備

内部副型はホストから送られ、ハードウェアによって消費され、それから適切な応答がホスト・ノードへ送り返される。内部パケットは媒体100へ決して送られない。したがって、このパケット型はペイロード・パケット節では定められておらず、PLX (内部) ホスト・パケットの下で定められた節にある。

## 【0249】

## 生ACK型

生ACK型はどの型の応答が現在の生データ・パケットにに続かなければならないかを指定する。応答型は4つの形式、すなわち、バースト (応答なし)、二重LOGI、LOGI、およびDACK、のうちの1つをとる。

## 【0250】

バースト型は説明を必要とせず、パケットは次々に送られる。バースト・シーケンスの最後のパケットには異なる確認応答型が割り当てられる (バースト・シーケンスを終了するために、応答が用いられる)。

## 【0251】

二重L o G Iシーケンスによってグループ要求すなわち放送要求を送ることが可能になる。ノードがパケットをバッファできない場合、それは最初の間隙間隔内で応答し、それがパケット正しく受信されてパケットを解析すると、遅延された間隙間隔中にそれは応答する。

#### 【0252】

L o G I応答は単一のノードへ向けられ、それは応答のための最も効率的な機構である。L o G Iパケットの長さは帯域幅に関して最も効果的であるが、応答についての非常に多くの情報を含むことはできない。

#### 【0253】

DACK応答は特定のノードへ向けられるが、応答内に、L o G I型よりもはるかに多くの情報を含むことができる。

#### 【0254】

<u>ビット (2、1)</u>	<u>パケット副型</u>
0 0	バースト
0 1	二重L o G I
1 0	L o G I
1 1	DACK

#### 【0255】

##### 暗号

暗号ビットによってパケットの内容を、認証バイトからスタートして、暗号化できる。1つの暗号化法が256ビットD i f f i e-H e l l m a nハンドシェイクを用いて鍵の変更をこなうことができ、その後で、秘密の32バイト・アレイが媒体100を通じて確実に送られる。その後のトランザクションは確実な通信のために暗号化アレイを使用できる。

#### 【0256】

##### ビット 3 : 暗号

現在のパケットが暗号化される=1

現在のパケットが暗号化されない=0

#### 【0257】

## ソケット

通常PLX生データ・ペイロード・パケットが下記のフィールド・サイズで構成される。

## 【0258】

<u>フィールド</u>	<u>長さ</u>
プリアンブル 1201	3バイト
長さ 1202、1203	2バイト 重複している
制御 1206	1バイト
宛先アドレス 1205	4バイト
発信元アドレス 1206	4バイト
ペイロード 1211	0～255バイト
CRC 1212	2バイト

同じノードに多数のアプリケーションが存在している場合、特定のノード・アドレス内の適切なアプリケーションへパケットを送れるようにする機構が用いられる。それらの型のアプリケーションはソケット・フィールドを使用する。第1のバイトは宛先ソケット・アドレスであり、第2のバイトは発信元ソケット・アドレスである。したがって、このビットを設定することにより、MACヘッダのサイズが2倍になる。このフィールドは、実施された時は、認証バイトの直後にあり、下記のビットがセットされると含まれる。

## 【0259】

<u>ビット</u>	<u>4</u>	<u>ソケット</u>
1		ソケット・フィールドを含む
0		ソケット・フィールドを含まない

## 【0260】

## プロトコルID (PID)

各パケットは、IPX、TCP/IP、またはCALなどのより高いレベルのプロトコルによって解析できる情報を含む。PLXは、ネットワークを横切って送信/受信すべきそれらの型のパケットを入れるトランスポートとして単に用いられる。通常は、より高いレベルの解析ルーチンがホスト・システムに存在する

が、最小セットのハードウェアはCAL解析機能を含むことが要求される。したがって、ハードウェアはCAL要求を解析し、適切なペイロード取扱いルーチンまでの他の全ての要求を送る。プロトコル情報の中にはハードウェア（たとえば、ROM、フラッシュメモリ等）内に配置できるものがあり、他のプロトコル情報はホスト・ノードにより解析される。このビットは、ハードウェア・プロトコル・ハンドラがこのパケットの解析を開始することを要求されているか否かを決定する。

### 【0261】

#### ビット 6      プロトコル ID (PID)

- 1                  プロトコルIDが存在する（マイクロ解析）  
 0                  プロトコルIDが存在しない（生-ホスト解析）

生パケットは、データの最初のバイトがその型のプロトコルのバイトコードではなく、その代わりにプロトコル・ヘッダ自身の最初のバイトであることを意味する。PID解析可能なパケットは最初のバイトコードを復号してどのプロトコルがそのパケットを解析すべきかを決定する。

### 【0262】

下記はPIDビットがセットされた時に利用できるオプションである。1番目のデータバイトは現在のパケットを解析するのに必要なプロトコルの型を表している。

<u>バイト値</u>	<u>定義</u>	<u>型</u>
0 x f f	予備	n / a
-----		
0 x f e	完成されたパケット	c e b u s R e s p
0 x f d	正しくないパケット	c e b u s R e s p
0 x f c	エラーパケット	c e b u s R e s p
-----		
0 x d f ~ 0 x f b	予備	n / a
-----		
0 x a 0 ~ 0 x d e	コンテキスト番号 (CAL)	c e b u s C m d

0x9f                      予備 (CAL)                      c e b u s C m d  
 0x00~0x9e              コンテキスト・クラス (CAL) c e b u s C m d

## 【0263】

宛先アドレス・フィールド

宛先アドレス1205は現在のパケットの宛先ノードを含む。

## 【0264】

あるノードがある要求を持っているか、他のノード要求に対して応答しているとき、それは、応答パケットの宛先のノードのアドレスを宛先アドレス・フィールド1205内に置く。そのノードがアクティブネットワーク・サーバまたはデータベース・サーバと通信できるだけであるならば、それはそのアドレスを宛先アドレス・フィールド1205内に置く。さもなければ、宛先アドレスは要求しているパケットの発信元アドレス・フィールド1206から通常得られる。

## 【0265】

あるPLXアドレスは周知である。それら周知のPLXアドレスのリストが以下に示す。

アドレス	摘要
0x00000000~0xffffffffef	有効で独自のノード・アドレス
0xfffffffff0~0xfffffffffc	予備
0xfffffffffd	アプリケーション・サーバ・ノード・アドレス
0xfffffffffe	アクティブネットワーク・サーバ・ノード・アドレス
0xffffffffff	ノード・アドレス放送

## 【0266】

発信元アドレス・フィールド

発信元アドレス1206は現在のパケットのノードのアドレスを含んでいる。ノードが要求を持っているか、他のノード要求に対して応答していると、それはそれ自身のノード・アドレスを発信元アドレス・フィールド1206内に置く。ノード・アドレスはノードの型と組合された8バイトGUIDの一部を用いて、4バイトのノード・アドレスを作成する。GUIDからの最下位から7桁目のニブル (Nibble) が用いられ、ノード型がノード・アドレスの最上位のニブル (8

番目のニブル) に重ね書きする。

【0267】

例：

If . . .

GUID=0x0123456789ABCDEF

And Node Type=0x03

Then . . .

Source Address=0x39ABCDEF

End If

【0268】

シーケンス番号フィールド

シーケンス・フィールド1207は、媒体100で伝送されるより小さいパケットに分けられたデータ・パケットまたはデータ・シーケンスを再構成すなわち再組立てする能力をホスト・アプリケーションに与える。二重シーケンス番号は捨て去ることができ、受信されなかったシーケンス番号は再び送ることができる。シーケンシングによってより大きいデータ流に対してデータを完全なものにする。シーケンス・フィールド1207内に置かれた値はアプリケーションに依存し、所望により他の目的に使用できる。

【0269】

認証フィールド

認証フィールド1208によって、各パケットの受信完了前に各パケットの有効性を確認できる。認証フィールド1208は、暗号化アレイの初めの2バイトに対して排他的オア操作を行うことによって通常シード (seed) される。したがって、安全システム内の全てのノードには同じ認証値がシードされ、それらの全てはこの検証手続きを受けなければならない。完全性を高めるために認証フィールドはさらに暗号化される。

【0270】

ペイロード・フィールド

データ・ペイロード・フィールド1211は受信ノードに情報を提供するため

に用いられる。ペイロード・データの第1のバイトは内容をどのようにして解析するかを決定するバイト・コードを含むことができる。データのこの最初のバイトは初めに説明した生ビットとともに用いられる。

#### 【0271】

周期的冗長検査 (CRC) フィールド

周期的冗長検査 (CRC) フィールド1212は、送信されたパケット内に信頼性のある誤り検出技術を用意するために用いられる。終了した時にそれは再評価され、認証のために比較される。この検査を通らなかったパケットは捨てられる。

#### 【0272】

CRCアルゴリズムは、不当なオーバーヘッド (ソフトウェアおよびハードウェアにおける) なしに、所望の信頼度レベルを得るように、十分効率的かつ簡単であるように選択される。送信されたパケットと受信されたパケットとに対する実行中のCRC計算を行えるに十分速いCRCアルゴリズムを提供することが望ましい。

#### 【0273】

実行中計算 (ビットまたはバイトが受信されたときに、全部のパケットが来ることを待っている代わりに、CRCが更新される。同じことが送信についてもあてはまる) は強制的ではないが、システムの全体のスループットおよび性能を高めるのに役立つ。

#### 【0274】

一実施形態では、 $G(x) = x^{16} + x^{15} + x^{11} + x^8 + x^6 + x^5 + x^4 + 3x + 1$  によって  $G(x)$  が与えられる。

#### 【0275】

PLX (内部) ホスト・パケット

PLX内部ホスト・パケットは媒体100には決して到達しないので、パケットについての説明ははるかに簡単にみえる。プリアンブル1201は必要とされず、また重複長さフィールド1202、1203と、アドレッシング・フィールド1205、1206も必要とされず、かつCRCフィールド1212も必要と



されない。図14は、長さフィールド1701と、制御フィールド1702と、データ・フィールド1703とを含むPLX内部ホスト・パケットのフォーマットを示す。データ・フィールド1703は制御フィールドが指定したいかなるものも含む。制御フィールドの以前の定義（それはPLX内部ホスト・パケットにも同様に適用される）に示されているように、ハードウェアとホスト・ノードの間を通り、トラフィックの流れを容易にするいくつかのパケットが存在する。以下は各型のパケットの定義である。

#### 【0276】

##### CAL要求パケット

図15はCAL要求パケット1800のフォーマットを示す。それは長さフィールド1701と、制御フィールド1702と、CALデータ・フィールド1803とを含む。制御フィールド1702は値0x11を有する。

#### 【0277】

CAL要求パケット1800は、ハードウェア上に存在するCAL情報を取ってくるために、ホストによってハードウェア・ノードへ送られる。PLXノードはアプリケーション・コードを有することができ、またはホスト・プロセッサがハードウェア/ASICから分離されているために、CAL情報はそれら2つの別々のプロセッサを越えて拡がることもできる。したがって、ホスト・プロセッサはCAL情報を取り付けられているノードから周期的に集める。

#### 【0278】

##### CAL応答パケット

図16はCAL応答パケット1900のフォーマットを示すものであつて、長さフィールド1701と、制御フィールド1702と、CAL応答フィールド1903とを含む。制御フィールド1702は値0x21を有する。

#### 【0279】

上で述べた理由と同じ理由から、CAL応答パケットはハードウェア・ノードから取り付けられているノードへ送られる。この応答パケット1900は先行するCAL要求パケット1500に応答して送られる。

#### 【0280】

### Tx状態パケット (単一チャネル、スピード)

図17は単一チャネルCAL応答パケット2000のフォーマットを示すものであって、長さフィールド1701と、制御フィールド1702と、データフィールド1903とを含む。制御フィールド1702は値0x21を有する。図18は多チャネルCAL応答パケット2100のフォーマットを示すものであって、長さフィールド1701と、制御フィールド1702と、データフィールド2103とを含む。制御フィールド1702は値0x31を有する。

#### 【0281】

Tx状態パケットの形式は2つある。1つの形式は単一チャネル、単一スピード・アプリケーションのためのものであって、制御バイト値0x21を用いる第2の形式は多チャネル、多スピード解決のためのものであって、0x31の制御バイトを使用する。

#### 【0282】

単一チャネル、単一スピード解決は使用できるただ2つのTxバッファを有し、それら2つのTxバッファの状態は内部PLXハンドシェイクによってホスト・ノードへ周期的に送り返される。それらのTx状態パケットの目的は、ホスト・ノードからハードウェアに渡された未解決の送信事象に関するループを閉じることである。しばしば、DACKパケット内の戻された同じ値が、この送信事象に関する情報のためにホストに渡されるが、多くの時にはDACKは外部PLX事象に渡され、その場合には、DACK値はホスト・ノードに渡してはならない。ホスト・ノードが送信要求を生じた時にDACK値はホスト・ノードに送り返される。

#### 【0283】

したがって、PLXは下に示す複製されたDACK状態値を使用する。

#### 【0284】

媒体上で見られるDACK状態フィールド値

0x0=バッファ・フル受信 (失敗)

0x2=ノードにより用いられるトークン (ホストへ送られない)

0x3=ノードにより用いられないトークン (ホストへ送られない)

0 x 4 = 「覚醒」 要求に応答するトークン (ホストへ送られない)

0 x 9 = プリンタ・シーケンス番号付けエラー

0 x a = プリンタ・プラグを引き抜かれたエラー

0 x b = プリンタ・オフラインエラー

0 x c = プリンタ一般的なエラー

0 x d = プリンタ紙エラーなし

0 x e = プリンタ不明エラー

0 x f = 成功

0 x 9 から 0 x e までの値はプリンタ・ノードからの DACK 応答である。プリンタ応答値はホスト・ノードへ修正されずに返される。

#### 【0285】

値 0 x f はうまくいった DACK 応答であって、ホストが要求を出ると、この値もホスト・ノードへ修正されずに返される。

#### 【0286】

値 0 x 2 から 0 x 4 は外部 PLX コマンド・パケットへの DACK 応答値であって、ホスト・ノードまで渡してはならない。

#### 【0287】

唯一の奇妙な状態値は 0 x 0 であり、これはワイヤ上で受信中のノードがビジーであり、パケットを受取ることができないことを意味する。ハードウェアはこの状況を認識し、指定された (それがビジーでない時よりもしばしば) このパケットを再び試みる。受信するノードが異常に長い時間ビジー状態に留まっていると、パケットは最後に中断され、「失敗 - 0 x f」応答状態がホスト・ノードへ送り返される。ホスト・ノードへ送り返された 0 x 0 の値は何も意味しない。それは完了しなかった送信事象のデフォルト値であり、ホストは非零状態がそのフィールド内に置かれるまで待つ。0 x 1 の値はワイヤに決して戻されない。ノードが間違ったデータを持つパケットをノードが受信すると、ノードは単にそのパケットに応答せず、送信ノードはそれを再送信することを要求される。送信パケットが時間切れとなりその最大再試行数になったとき 0 x 1 の値がホストに戻されるだけである。

## 【0288】

下記は、ホスト・ノードへ通常戻されるTx状態値を示す表である（値はあらゆる場合にDACK応答値と同一でないことに注意されたい）。

## 【0289】

Tx状態データ・フィールド値

- 0x0 = このTxバッファのためのTx状態なし
- 0x1 = 失敗（Tx時間切れまたは受信バッファ一杯）
- 0x9 = プリンタ・シーケンス番号付けエラー
- 0xa = プリンタ・プラグを引き抜かれたエラー
- 0xb = プリンタ・オフラインエラー
- 0xc = プリンタ一般的なエラー
- 0xd = プリンタ紙エラーなし
- 0xe = プリンタの未知のエラー
- 0xf = 成功

これは続くDACK情報が内部Tx状態パケットを介してホスト・ノードまで渡されないことを意味する。

## 【0290】

ホストに与えられていない追加のTx状態Info

- 0x0 = 受信バッファ一杯（失敗）
- 0x2 = ノードによって用いられるトークン（ホストへ送られない）
- 0x3 = ノードによって用いられないトークン（ホストへ送られない）
- 0x4 = 覚醒要求に応答するトークン（ホストへ送られない）

Tx状態バイトは各部分がニブル幅である2つの部分にさらに分けられ、2つのTxバッファ状態を現す。Tx状態フィールド中の値をそれぞれの意味と共に以下に示す。

## 【0291】

Tx状態値の例

- 0x0f = 送られることに成功した最初のTxバッファ
- 0x0f = 送られることに成功した2且目のTxバッファ

0 x f f = 両方のT xバッファが送られることに成功した

0 x 1 f = 2番目のT xバッファは失敗し、最初のT xバッファは成功  
等. . .

#### 【0292】

T x状態パケット (多チャンネル、スピード)

T x状態パケットの第2の形式は多チャンネル、多スピードの解決策のためのものである。単一チャンネルT x状態パケットに関する以前の説明の全体と、それがDACK値にどのように関連するかは、いぜんとしてあてはまる。違いは、多チャンネル/スピードT x状態パケット内に含まれているデータ情報の量である。そのパケットは存在する各チャンネルに対して以前に定められた単一の状態バイトを基本的に含む。結果は各バイトが2つの別々のT xバッファを有する単一のチャンネルを表している多数のデータ・バイトである。

#### 【0293】

パケット・タイミング、間隔取りおよび再試行

媒体100上で送信のために提示された全てのパケットは厳密なタイミング要件を守らなければならない。それらのタイミング要件は、システムが円滑かつ衝突なしに動作できるようにする諸規則である。それらの規則を守ることは正しい動作のために厳密に実行されねばならない。

#### 【0294】

正常な動作の下では、「アクティブネットワーク・サーバ」がシステム上に存在し、媒体100をアクセスするためにアクティブノードの全てを調停しなければならない。以下の仮定は媒体100に存在するそのようなアクティブ状態に適用される。媒体100上でのインアクティブは、各ノードが睡眠状態にあり、したがって、「アクティブネットワーク・サーバ」として主張する前に正常な「聴取」プロセスを通らなければならない。

#### 【0295】

さらに、PLXシステムは確認応答されるハンドシェイク・シーケンスを特徴とする。確認応答パケットは指定された時間間隔内に戻されるようになっている。確認応答(DACK、LOGI、または二重LOGI)パケット以外のいかな

るものでも送信する前にトークン・パケットは要求される。アクティブネットワーク・サーバはトークン、またはL I Pパケットを送信する権利を有するノードのみである。クライアント・ノードはペイロード・パケットおよび確認応答パケットを送信するのみである。

### 【0296】

#### 典型的なパケット・タイミング

図19はパケット・タイミングと間隔決定を示すタイミング図である。パケット時刻は第1の基準時刻2202および第2の基準時刻2204に関連して定められる。第2の基準時刻は、 $50\mu s$ （マイクロ秒）の平均パケット間間隙（I / G a p）だけ第1の基準時刻に遅れている。

### 【0297】

上で示した図は650kbpsで動作するシステムのためのタイミングをとっている。間隙間タイミング以外の全ての値は表A4に与えられているようにして調整すべきである。表A4では上付き数字1は第1の基準2202を基準にした時間を示し、上付き数字2は第2の基準2204を基準にした時間を示す。

### 【0298】

	350kbps	700kbps	1.2mbps	1.4mbps
Min I / Gap1	$15\mu s$	$15\mu s$	$15\mu s$	$15\mu s$
AvgI / Gap1	$50\mu s$	$50\mu s$	$50\mu s$	$50\mu s$
プリアンプル	$130\mu s$	$65\mu s$	$38\mu s$	$33\mu s$
loGI Packet2	$140\mu s$	$70\mu s$	$40\mu s$	$35\mu s$
DLogI Packet2	$185\mu s$	$92\mu s$	$54\mu s$	$46\mu s$
DACK Packet2	$335\mu s$	$168\mu s$	$98\mu s$	$84\mu s$
TxRetry LoGI1	$205\mu s$	$103\mu s$	$61\mu s$	$52\mu s$
TxRetry DACK1	$400\mu s$	$200\mu s$	$117\mu s$	$100\mu s$
TxRetry LoGI1	$320\mu s$	$160\mu s$	$94\mu s$	$80\mu s$
インタートークン1	$3+ms$	$3+ms$	$3+ms$	$3+ms$

表A4. パケット・タイミング

### 【0299】

正常な条件の下では、典型的なパケット・タイミングは、パケットを受信しているノードを所定の長さの時間内に応答することを必要とする。この応答時間は、L o G I / 二重L o G I 確認応答パケットを除く全てのパケットで同じである。したがって、パケット・タイミングの2つのケースは1) L o G I / 二重L o G I シーケンスと2) 他の全ての応答である。

### 【0300】

#### 他のパケット・タイミング

ノードは、応答パケットを要しないバースト・パケットおよび確認応答パケットを除くパケットを、ペイロード・パケットを生じたノードへ、ある時間内に、送り返す。応答パケットは型：DACKパケット、L o G I パケット、またはペイロード・パケットにできる。

### 【0301】

応答パケットは上で図19に示されている間隙間隔要求に従う。最短応答時間は通常5マイクロ秒より長く、最長応答時間は通常50マイクロ秒を超えてはならない。

### 【0302】

送信ノードが前回の送信の確認応答を受信しないと、配信信頼度を高くするために再試行プロセスを開始しなければならない。この再試行プロセスは最長の可能な確認応答シーケンスまたはDACKパケットの長さプラス最も広い可能な間隙間隔の後で、または350 k b p s で約400マイクロ秒の後で通常始まる。

### 【0303】

#### ノードに特有の情報

各ノードはその特定のノードを特徴付けるある量の情報をもって構成されるようになる。PLXノードはシステムで完全に機能するためにこの最少量の情報を要する。

### 【0304】

固有の識別、アドレス指定可能性、および大域的に固有な識別 (G U I D)

PLXノードが電気装置に差し込まれると、そのノードはただちに動作できる

状態になる。各ノードは一連番号が焼き付けられて来る。その一連番号のうちの  
下位の28ビットはそのノードの実行時間アドレスとして使用される。これは大  
域的な固有性を確保せず、衝突するアドレスを有する2つのノードを見付ける貴  
方の機会が2億6千8百万に1回であるので、それは衝突の可能性を制限する。  
このより大きい実行時間アドレスはスループットを僅かに減少させるが、システ  
ムを簡単にして（ノードは工場から予め組み立てられて出荷されるためである）  
、プラグアンドプレイ性能を高くし、かつ使用を容易にする。

### 【0305】

ユニバーサル・コンテキストおよびノード輪郭オブジェクト

CEBus／一般的CALコンプライアント・ノードは、最低限、ユニバーサ  
ル・コンテキストおよびノード制御オブジェクトを有し、それに変数例（IVs）  
）が組合わされている。PLXはCEBus／一般的CALが定めた報告条件お  
よびアドレス指定からそれる（両方とも、CEBus／一般的CALピアツーピ  
アアーキテクチャとは異なって、PLXクライアント／サーバ・アーキテクチャ  
に関連している）。したがって、PLXはユニバーサル・コンテキスト／ノード  
制御オブジェクトを、IV記述が僅かに異なるノード輪郭オブジェクトとして再  
構成する。また、各PLXコンプライアント・ノードはノード制御オブジェクト  
に組合わされた変数例を含む。

### 【0306】

各ノードは、識別する所定の属性セットを含む責任を負い、一般に知られてい  
る属性を持つノード型群内にノードを置く。各ノードに対するノード輪郭オブジ  
ェクト情報はノード内の不揮発性メモリにハードコードされる（hard-cod  
ed）ことが好ましい。ノード輪郭オブジェクトは変数例のリストで構成され  
ている。各PLXノードは、下の表A5に示されているユニバーサル・コンテキ  
スト（0x00）と、ノード輪郭オブジェクト（0x01）と、指定された変数  
例（IV）とを少なくとも含む（ここにR/Wは読出し／書込みを示す）。

### 【0307】

【表1】



IV	R/W	型	名称	記述
o	R/W	d	context_list	特定のノードによりサポートされている全てのコンテキストのリストを含む
w	R/W	b	power	この特定のノードへの大域電力を制御
s	R	d	serial_number	製造者が割り当てた製品一連番号を含む。その下位8バイトは装置GUID（大域的に独特な識別）（18バイト）でもある。
n	R	c	manufacturer_name	製造者特定名（最大18バイト）
m	R	c	manufacturer_model	製造者特定モデル（最大18バイト）
c	R	n	product_class	一般的GAL仕様（2ASC 11バイト） ごとについて
v	R	c	conformance_level	この特定の装置CAL/PLXサポートの現在のレベルを特定するストリング（4ASC 11バイト）
h	R/W	d	area_address	経路指定およびネットワーク特定のために用いられる（1バイト）。このIVは常に大域的に読出しできる（network_nameとともに）。
a	R/W	d	unit_address	直接アドレスされるパケットのために用いられるID（4バイト）
i	R	d	network_class	<p>装置のネットワーク・クラスを定め、装置MACアドレスの最上位のニブルを上書きするために用いられて、トークンの支出に優先順位をつける。下記は優先度とネットワーク型に関連する値である。</p> <p>0x01 ビデオ・システムⅠ  0x02 ビデオ・システムⅡ  0x03 オーディオ・システムⅠ  0x04 オーディオ・システムⅡ  0x05 予備  0x06 安全システム  0x07 ユーティリティ監視システム  0x08 HVACシステム  0x09 照明装置  0x0a 機器システム  0x0b データ・ネットワーク化システム  0x0c 予備  0x0d 予備  0x0e 予備  0x0f 大域システム</p>
f	R	d	buffering	受信バッファのサイズ（バイトで）
x	R	c	product_rev	製品改進レベル（4ASC 11バイト）
b	R/W	d	dynamic_mask	<p>単一ビットにより特徴づけることができるあるダイナミック・ノード機能を含む。</p> <p>ビット0：無差別モード  1 = イネーブル  0 = ディセーブル  ビット1：MACサーバ  1 = MACサーバ  0 = 非MACサーバ  ビット2：規則/データベース・サーバ  1 = データベース・サーバ  0 = 非データベース・サーバ  ビット3：インアクティブ/アクティブ  （ポーリングされた・ポーリング）  1 = 現在アクティブ  0 = 現在インアクティブ</p>

【表2】

u	R	d	static_mask	<p>単一ビットにより特徴づけることができるある静的ノード機能を含む。</p> <p>ビット0: 遠隔明滅可能性 1 = 遠隔可能 0 = 遠隔不能</p> <p>ビット1: 認証可能 1 = 認証可能 (NVMemを要す) 0 = 認証不可能</p> <p>ビット2: 複雑な方法サポート 1 = 複雑な方法サポートされる 0 = 複雑な方法サポートされない</p> <p>ビット3: Diffie/Hellman Maxキーサイズ 1 = 512ビット 0 = 256ビット</p>
y	R	d	statistics	<p>下記のフォーマットでこのノードにより保持されている全ての関連するカウンタの統計表</p> <p>バイト0: 表バージョン バイト1: ビットマスクカウンタ</p>
r	R/W	d	reset	このノードをリセットできるようにする。0x521Aの値をこの変数例に書いてリセット機能を開始する。
l	R/W	b	sleep	サービスまたは手動制御のためにオンラインまたはオフラインへノードが入れるようにする。
G	R/W	d	group_address_list	このノードによりサポートされている全ての群アドレスの長さが先行したリスト。したがって、最初の16ビット値は後続する群アドレスの数である。
j i k g d	R/W	d	authentication	<p>構成および初期化時にノードに対して解析されたID値の認証</p> <p>Diffie - Hellman中にXORされた暗号アレイ</p> <p>Diffie - Hellman中にXORされなかった暗号アレイ</p> <p>公衆鍵 公衆ゼネレータ 乱数</p>
q	R/W	c	network_name	理解可能な名称を持つユーザーにより指定された安全ネットワーク内にノードを置くことができるようにする。
e	R/W	c	product_name	論理名でノードを参照できるようにする。
p	R/W	c	product_location	理解可能な名称を持つユーザーにより指定された場所にノードを置くことができるようにする。
	R/W	d	*system_id_list	この環境領域内の全ての割り当てられたシステムIDのリスト
	R/W	c	*last_log	最後に記録された事象

表 A 5

下の表A 6は記憶されて、管理され、「アプリケーション・サーバ」により保持されて、アプリケーション・サーバ内のデータベースに存在するクライアント I V sを示すものである。したがって、クライアントはそれらの I V sに関する情報の蓄積および提供に関わる必要はない。

## 【0309】

また、マスタ・ケースのみに対するユニバーサル・コンテキストの部分は、C  
ALにより定められたデータ・メモリ・オブジェクトを使用する規則オブジェク  
ト（0x03）、および我々の目的のために定められたある固有のIVsである  
。下記はこの特定のオブジェクトの記述である。

## 【0310】

【表3】

IV	R/W	型	名称	例変数記述
r	R/W	d	current_rule	現在の指標変数により指されたアクティブな規則を含む。
C	R/W	n	current_index	現在の規則変数に示されている規則の指標（ハンドル）を含む。
s	R	n	Rule_length	現在の規則変数に示されている規則の長さを含む。
m	R	n	Maximum_index	現在のインデックス変数内に置くことができる最大インデックス値を含む（最小値は常にゼロ）。
p	R/W	d	Previous_value	対応する規則内の記述された各IVの以前の値の列を含む。各IVは長さが先行されて零終了／零が結められている。
l	R	n	Previous_value_length	previous_value列の最大長さ
n	R/W	c	Rule_name	この特定の規則に割り当てられた論理名。ユーザーインタフェースをいっそう読みやすくするために用いられる。
z	R/W	n	Status	現在の規則の状態を含む。それが零であれば、この規則が適用され、もし非零であれば、この規則により向けられているIVの1つがインアクティブである（オフライン）。

表A6

規則オブジェクトによって遠隔ノードを、加え（跡継ぎ）、削除（跡継ぎしな  
い）、および規則リスト内の規則を見る（アレイを獲得する）ことができるよう  
にされる。

## 【0311】

ユニバーサル・コンテキストを提供することにより、ネットワークはノード・

リストを含むことができる。ノードはコンテキスト・リストを含むことができる。ノードは各コンテキストごとにオブジェクト・リストを有することができる。オブジェクト・リストが与えられると、ノードは特定の例変数を含むこともできる。それらのリストの多くは一般的なCAL仕様（ネットワーク・リストおよびノード・リスト以外）内で指定される。

### 【0312】

要求されると、ノードは、ノードの特定の構成について上で示したノード輪郭の特定の部分で応答する。ノード輪郭によって、当該ネットワーク内で固有である特定のノードを自動構成する手段を得ることができる。重複ノードが、一意に特定されるようにするために、他のレベルの構成を行うことができる。

### 【0313】

#### 安全

安全は2段階法により実現される。最初に、ネットワーク上で電源を投入された各ノードが公衆ネットワーク内に直ちに置かれる。公衆ネットワークは全てのノードに対するデフォルト・ネットワーク割り当てであって、他の全ての公衆ノードが利用でき、その認証IDがNULLに割り当てられる。下記の鍵交換プロセスによってノードが安全になったら、その認証IDは暗号化アレイにより指示された値に変化する。各ノードがこの私用／安全ネットワークに割り当てられると、それらに32バイトの暗号化アレイが与えられ、それから以後のパケットを暗号化し、または解読する。これはDiffie-Hellmanとして知られている鍵交換技術で行われる。効率的な指数関数的アルゴリズムを用いると、鍵交換に用いられている値の計算に必要な時間が減る。暗号化アレイがネットワークの各ノードのメモリ内にひとたび記憶されると、暗号化と解読とが行われる。一実施形態では、暗号化と解読は、帰還を含む排他的論理和を基にした流れ暗号化技術を使用する。たとえば、DES、RC4、MD5等を含む他のアルゴリズムを使用できる。

### 【0314】

#### 追加の諸特徴

#### 報告条件仕様

PLXの下では報告条件はそれがCALの下におけるものとは異なったやり方で取り扱われるので、規則を取り扱うPLX法をここで示すことにする。それらの交換は、厳密なCAL報告条件方法内で固有の制約の多くをアドレスするために実行された。違いを下の表A7で示す。

## 【0315】

<u>CEbus</u> <u>CAL</u>	<u>PLX</u>
1   オブジェクト当りの規則	オブジェクト当りの多くの規則
1   オブジェクト当りのアクティブIV	オブジェクト当りのアクティブな多数のIVS
簡単な規則のみ	簡単な規則および複雑な規則
堅い規則	融通性のある規則

表A7. 一般的なCALに対するPLXの諸利点

## 【0316】

一般的なCALの下における分布された規則とは逆に、PLX規則はサーバに存在しているので、PLXの単一の強力なエンジンによってPLXはそれが諸規則をどのように取り扱うかにおいてより強力である。これはどのようなIV変更でもその通りである。サーバがIV変更を見ると、サーバは変更された特定のオブジェクト／IV組合わせを調べ、サーバはその規則リストを調べ、そしてサーバは各規則の有効性を試験する。したがって、下記の2つのIVsを含むように各オブジェクトは構成され、それは作成された各規則を、指定されたオブジェクトおよび下に掲げた関連するIVに対して取り扱う。

## 【0317】

IV   R/W   型   名称	コンテキスト機能
R   R/W   d   rules__array	マスターズ規則リスト（規則オブジェクト）内にインデックスのポインタのアレイを含む。 各エントリは、このオブジェク

ト内のIVが変更された時に  
完全な規則を試験することを  
意味する。

P R/W n number of rules 規則アレイ内の規則  
の数を含む。

### 【0318】

実際のreport\_header、およびreport\_address、  
report\_condition、およびprevious\_value変数は  
、アレイにより指された規則内におおの保持される。読出しルーチンはこのポ  
インタ（またはインデックス）を単に規則エンジンへ送り、規則エンジンはそれ  
が必要とする適切な情報をマスタ規則リストから分析する。

### 【0319】

不揮発性メモリの取扱い

各ノードはROMなどの静的メモリ場所にノードプロファイル情報を含む。ま  
た、ノードは認証鍵などのその他の情報を不揮発性メモリに記憶できるが、これ  
は任意のものであって、いずれのPLXコンプライアント・ノードでも要求され  
るものではない。他の任意のメモリ要求は経路指定情報とその他のダイナミック  
な表を含む。

### 【0320】

クライアント変更通知

クライアント・ノードは状態変化条件をアプリケーション・サーバに通常知ら  
せる。これは、アプリケーション・サーバがその状態を変更することをクライ  
アントに知らせるとしても、クライアントはアプリケーション・サーバにその  
状態が変化したことを逆に知らせることを意味する。これによってアプリケーシ  
ョン・サーバのデータベースが実際のクライアント・ノードの変数に同期されて  
いないというような問題が起きる機会が減少する。

### 【0321】

これは、アプリケーション・サーバが通知条件およびクライアント変数変化に

関連する規則を含んでいるために望ましい。クライアントはこれに関しては知的ではないので、クライアントはアプリケーション・サーバに適切な変更を通知すべきである。

### 【0322】

アプリケーション・サーバは、クライアントが状態を変更したことを「アプリケーション・サーバ」に通知する、そのクライアント・ノードから確証を受けとった後まで、特定のクライアント・ノードに属するそのデータベース変数を通常更新しない。

### 【図面の簡単な説明】

#### 【図1】

パーソナル・コンピュータなどのスマートなノードと、外部保安灯などのダムなノードを有するネットワークのブロック図である。

#### 【図2】

7層OS I ネットワーク・モデルのブロック図である。

#### 【図3】

ユニバーサル・ゲットウェイ構造のブロック図である。

#### 【図4】

サーバアクティブ アルゴリズムの流れ図である。

#### 【図5】

規則コンポーネントを示すデータ図である。

#### 【図6】

スマート装置用のPLXネットワークモデルのブロック図である。

#### 【図7】

ダム装置用のPLXネットワークモデルのブロック図である。

#### 【図8】

媒体アクセスのアルゴリズムを示す流れ図である。

#### 【図9A】

アクティブネットワーク・サーバ・スピitting・アルゴリズムを示す流れ図である。

## 【図9B】

クライアントスピitting・アルゴリズムを示す流れ図である。

## 【図10】

アクティブネットワーク・サーバ・ポーリング・アルゴリズムを示す流れ図である。

## 【図11】

PLX論理群分離 (LOGI) パケットのフィールドを示すブロック図である。

## 【図12】

PLX生データ・パケットのフィールドを示すブロック図である。

## 【図13】

PLXトークン・パケットのフィールドを示すブロック図である。

## 【図14】

PLX直接確認応答 (DACK) パケットのフィールドを示すブロック図である。

## 【図15】

PLXマスクされたラインアップ挿入 (LIPG) パケットのフィールドを示すブロック図である。

## 【図16】

PLX直接ラインアップ挿入 (LIPD) パケットのフィールドを示すブロック図である。

## 【図17】

PLX内部ホスト・パケットのフィールドを示すブロック図である。

## 【図18】

PLX共通アプリケーション言語 (CAL) 要求パケットを示すブロック図である。

## 【図19】

PLX CAL応答パケットを示すブロック図である。

## 【図20】



PLX単一チャネル送信状態パケットを示すブロック図である。

【図21】

PLX多チャネル送信状態パケットを示すブロック図である。

【図22】

PLXパケット・タイミングを示すブロック図である。

【図1】

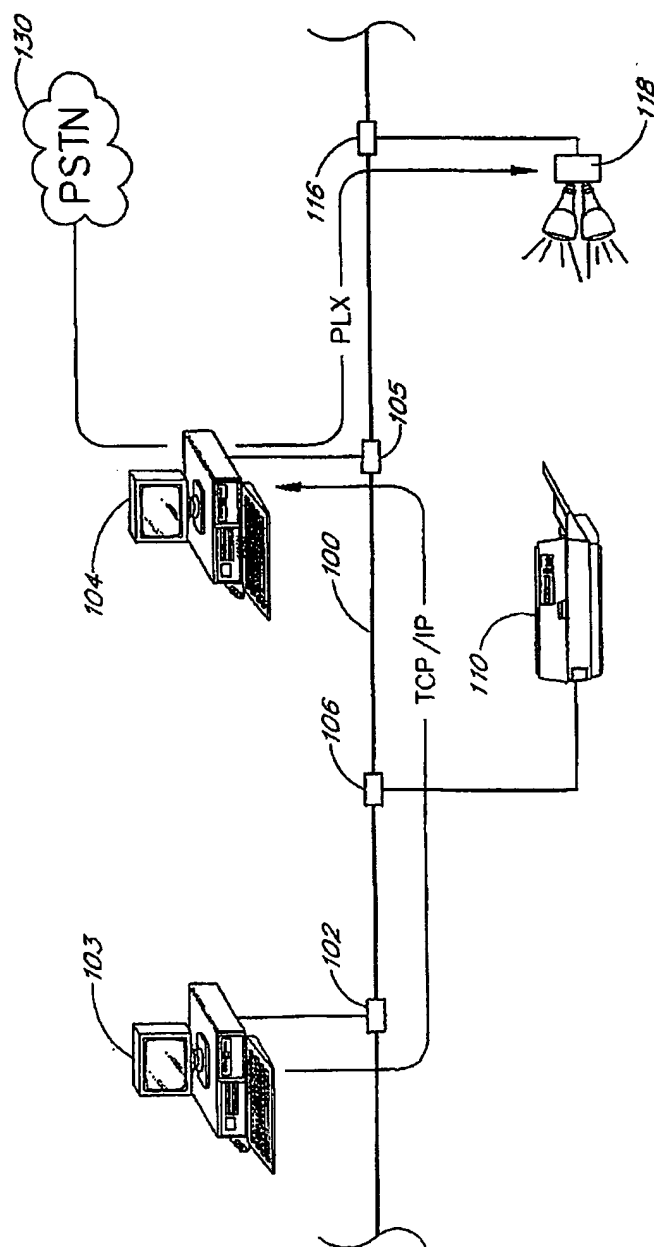
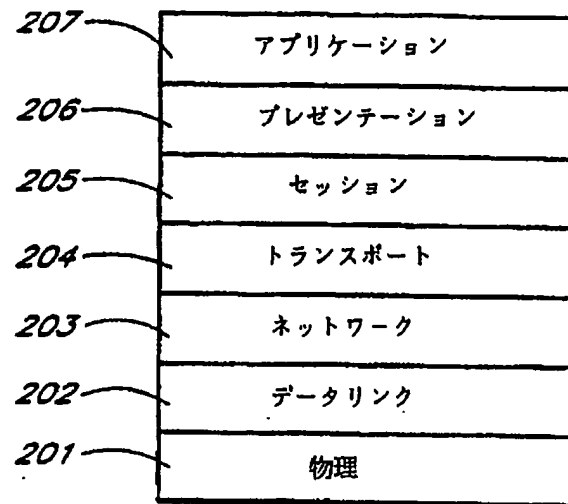


FIG. 1

【図2】

典型的なネットワーク化された  
PC（データ装置）



【図3】

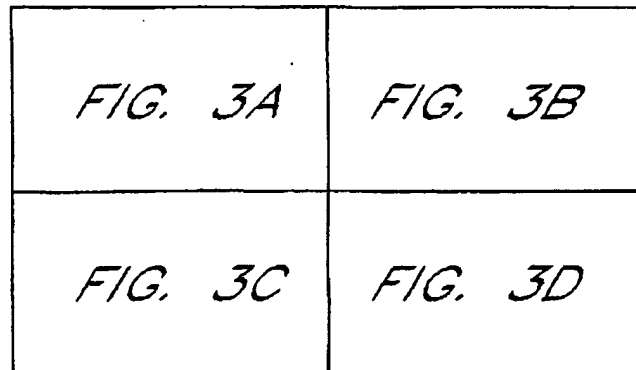
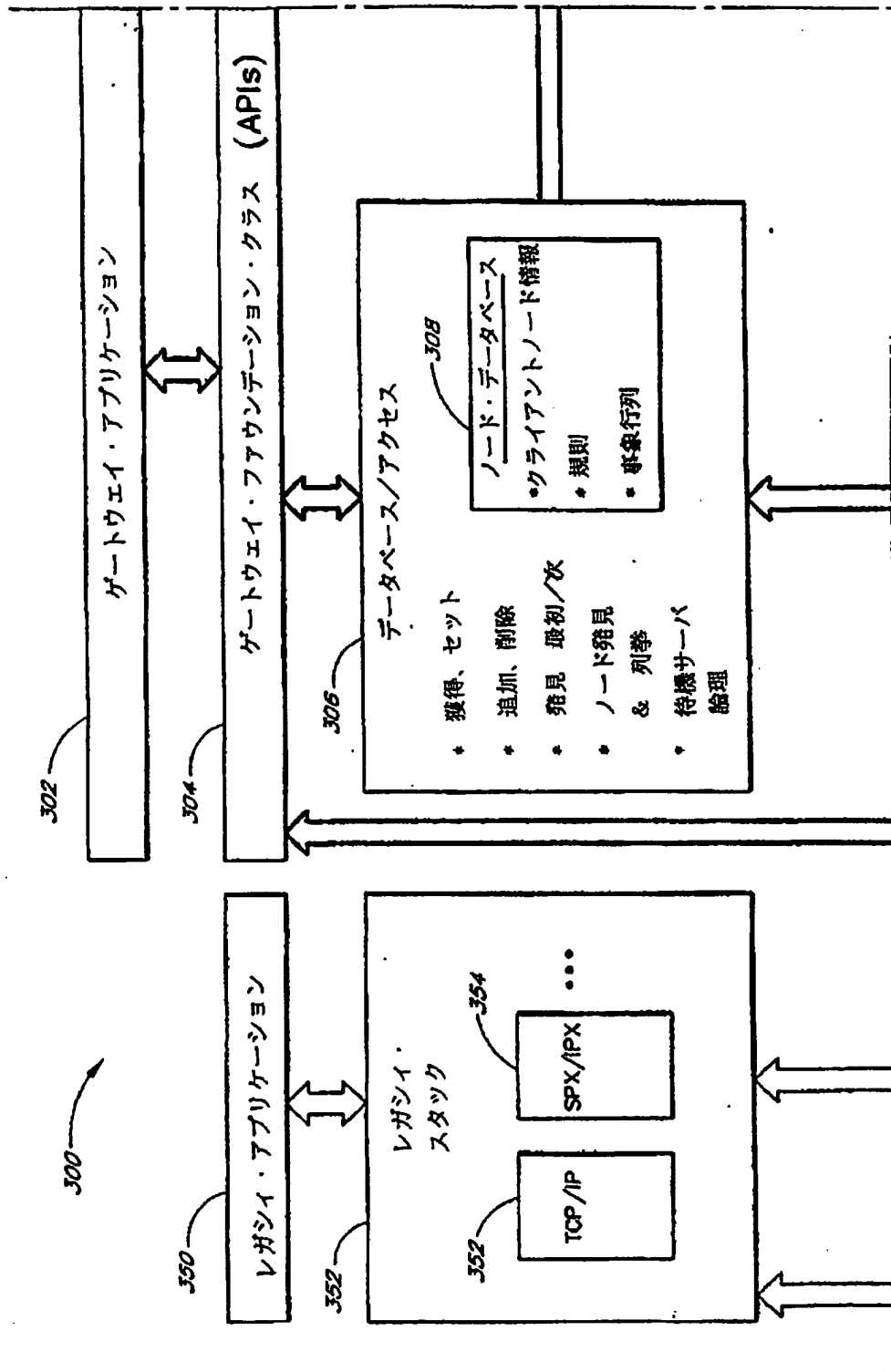
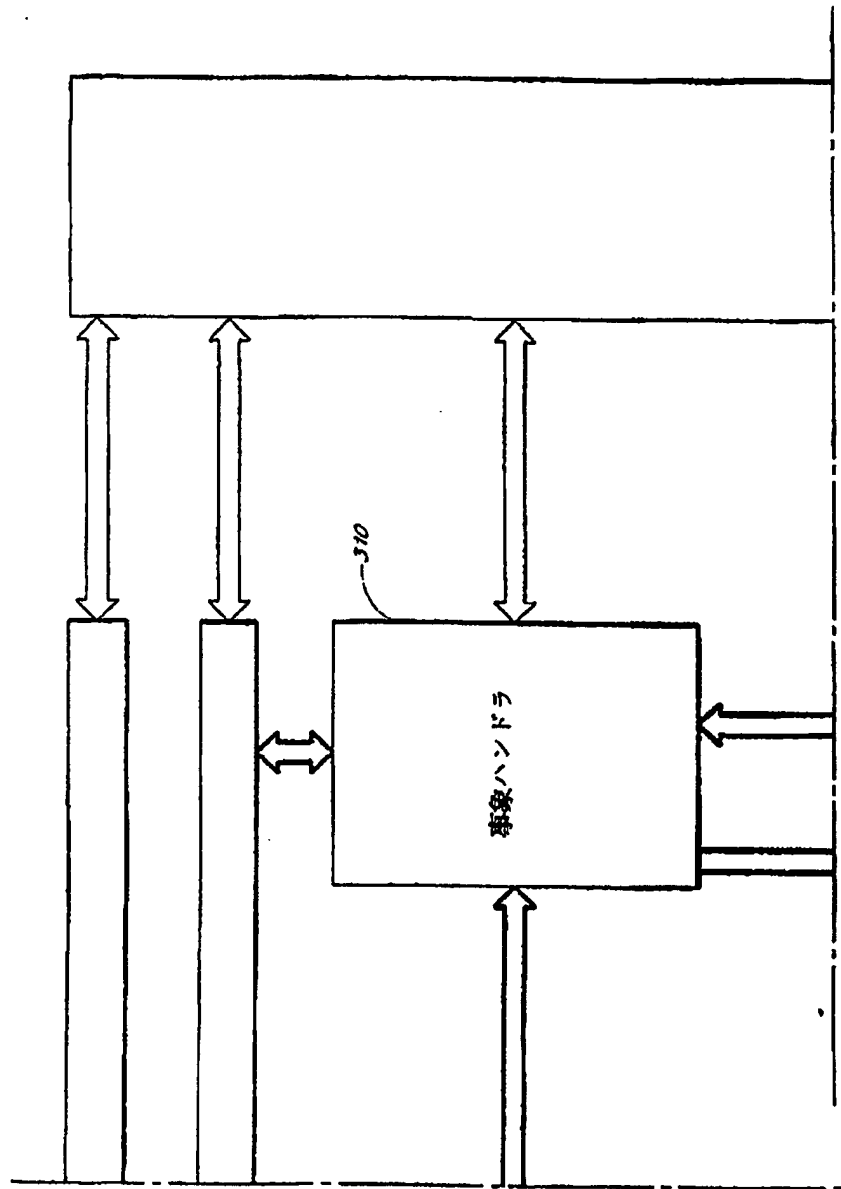


FIG. 3

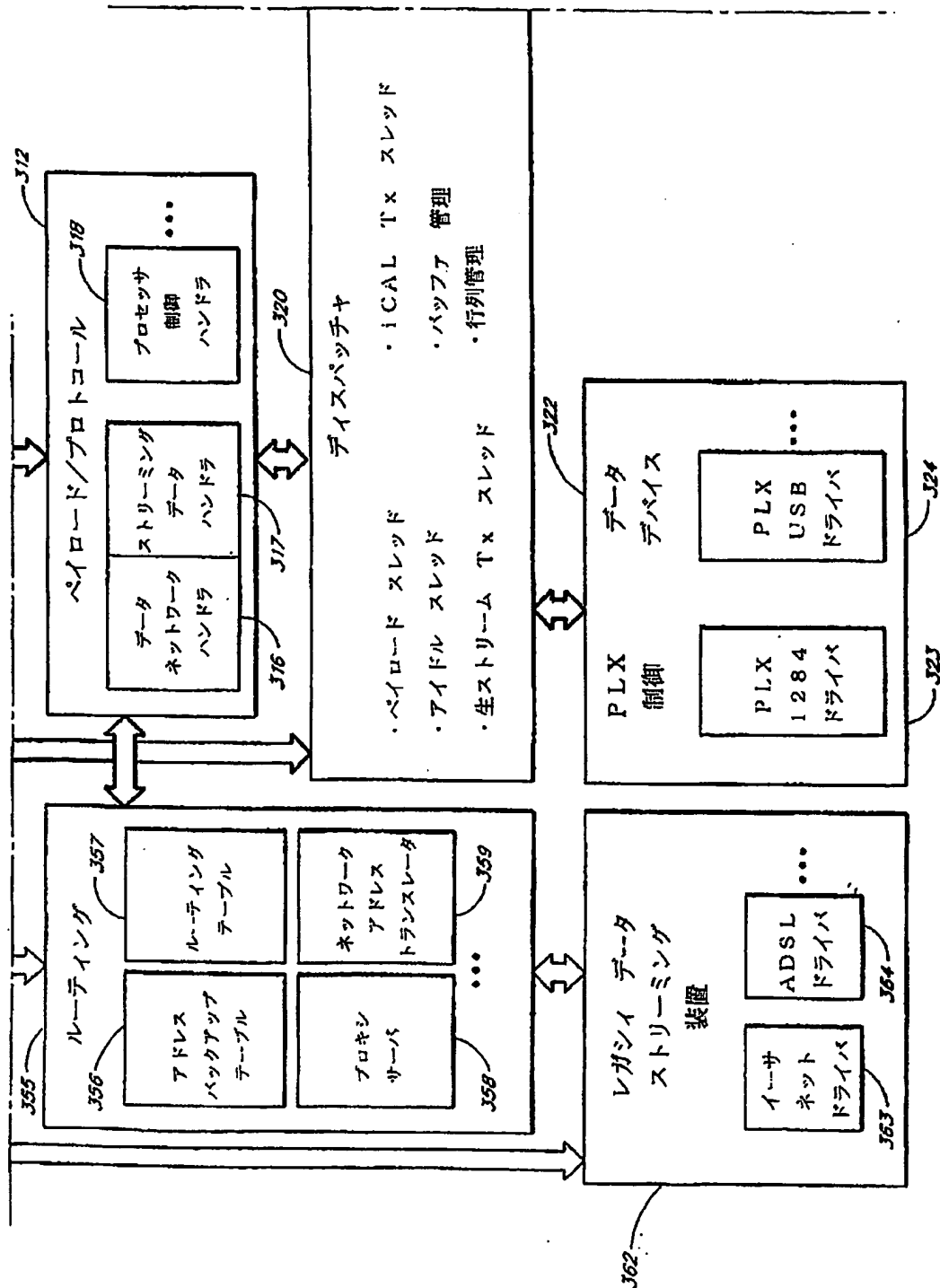
【図3A】



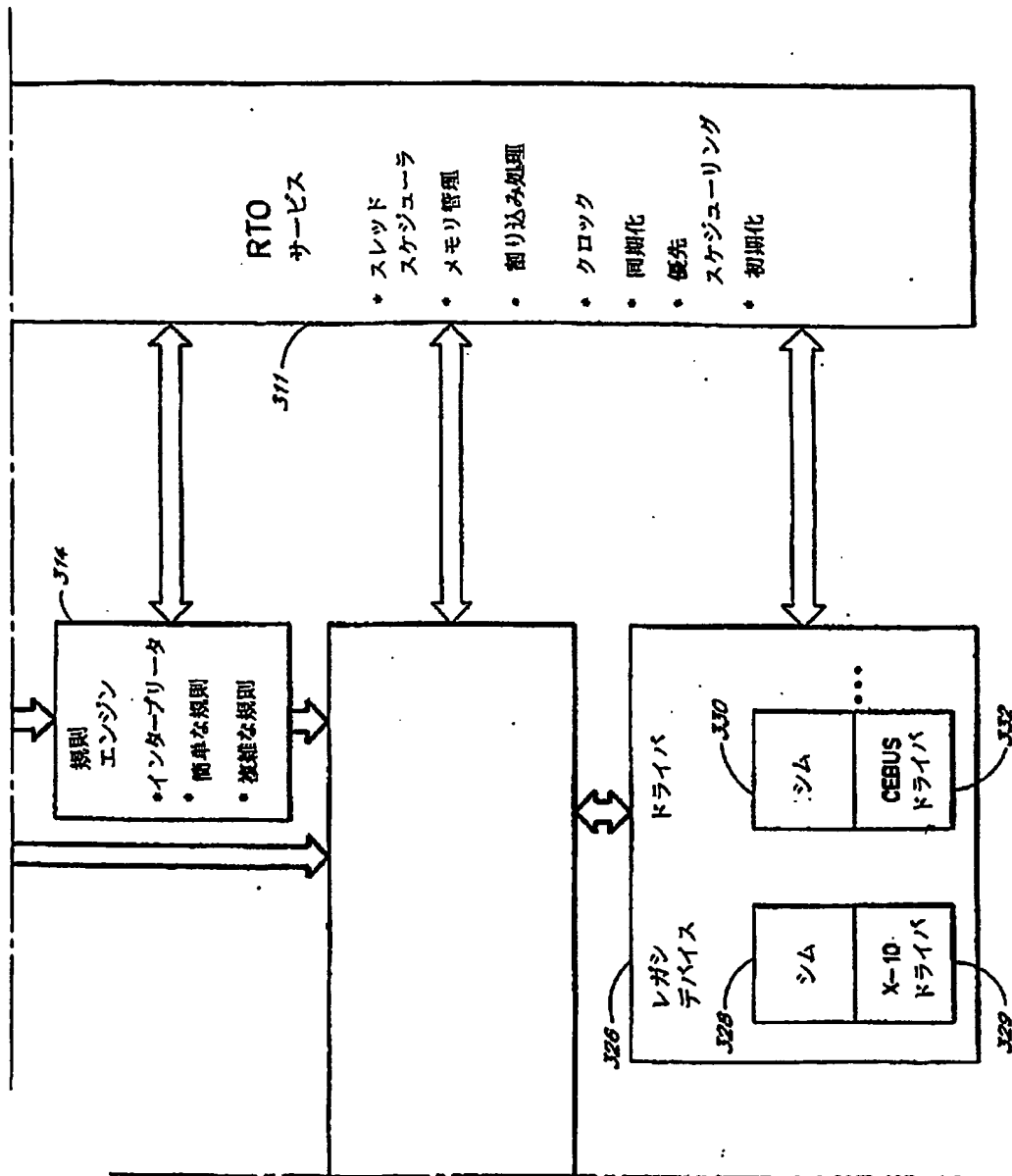
【図3B】



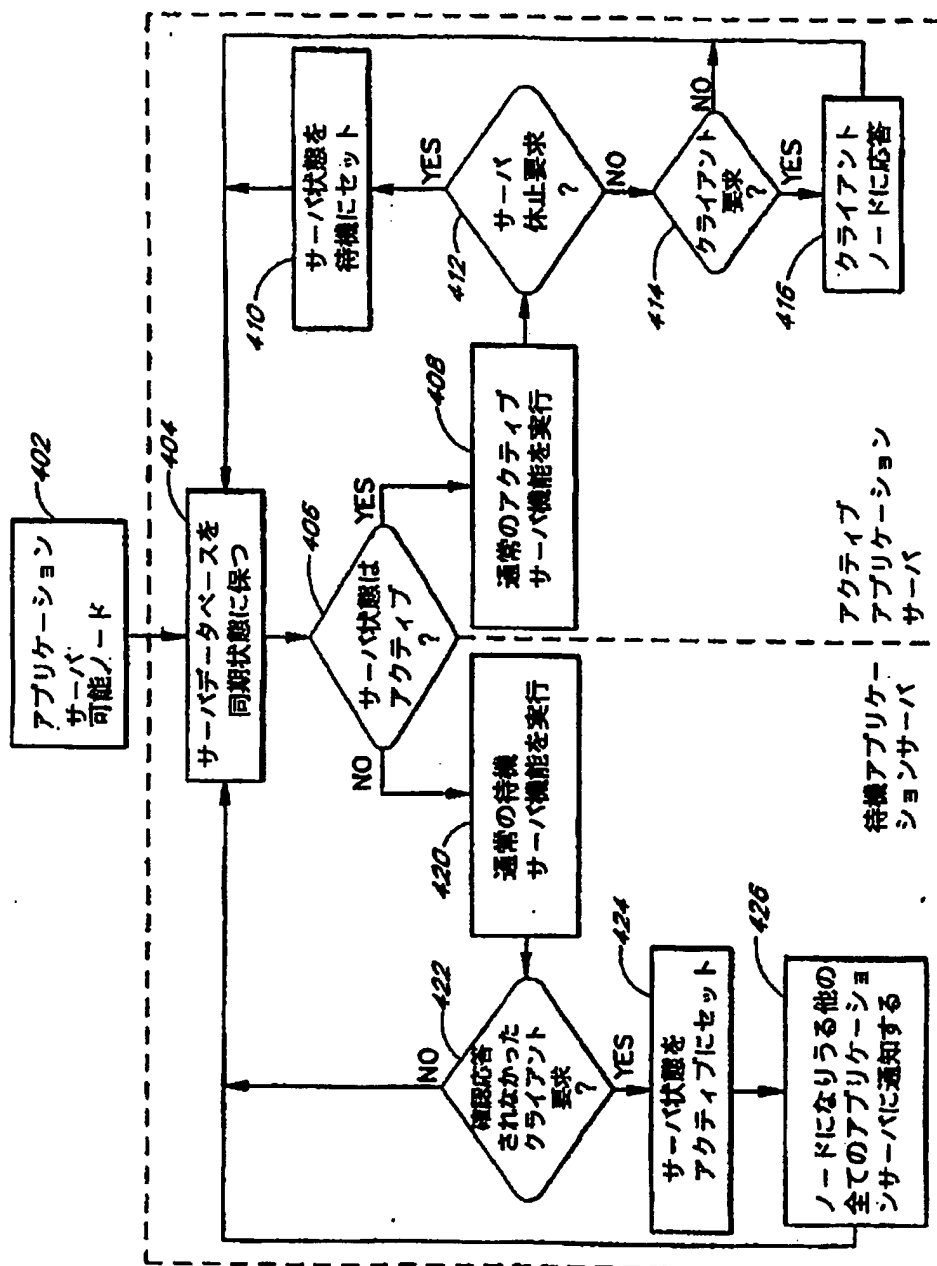
【図3C】



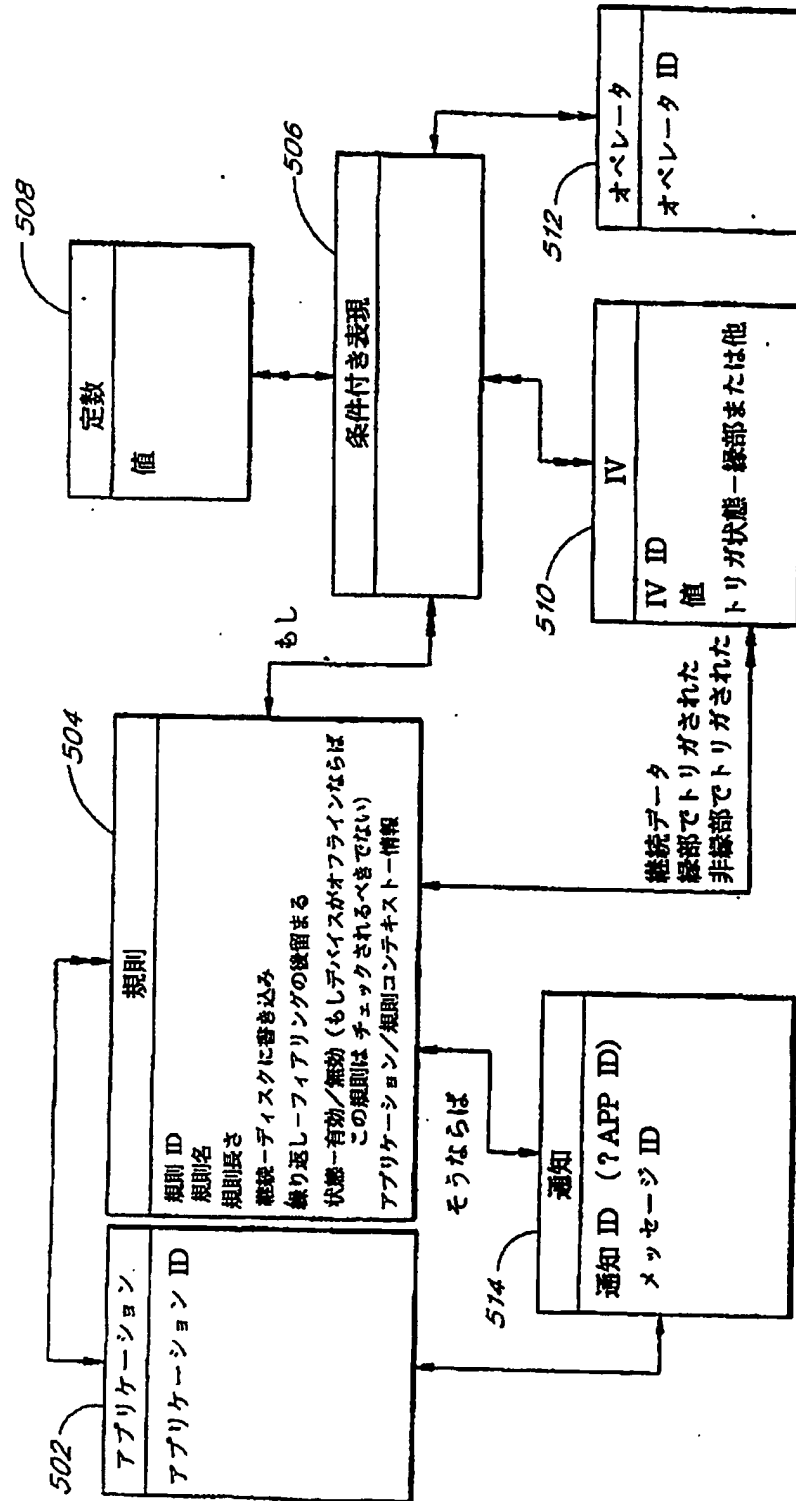
【図3D】



【図4】

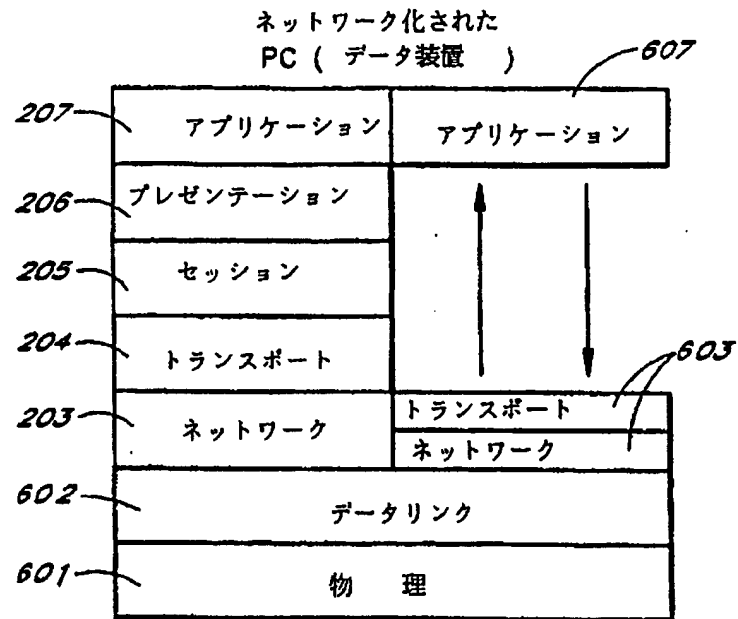


【図5】

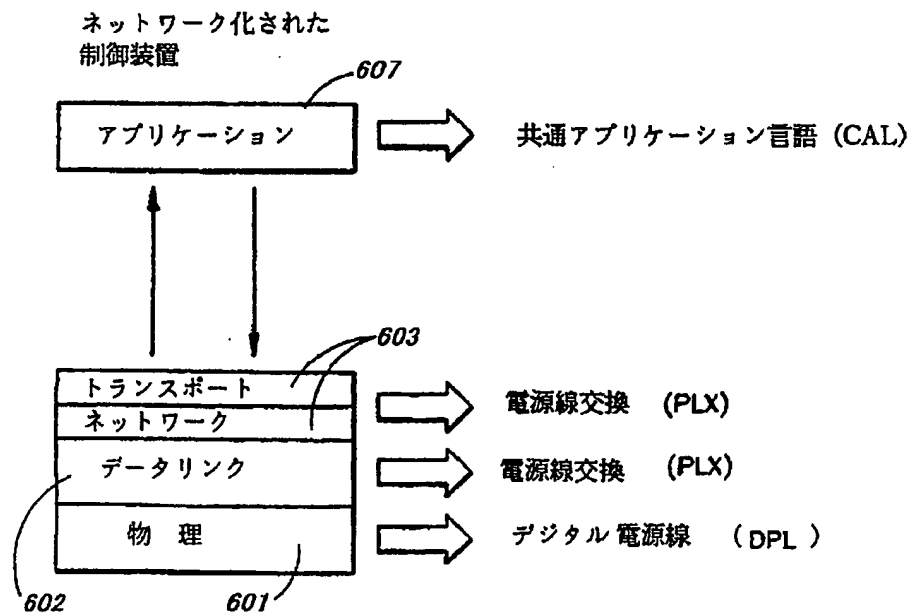




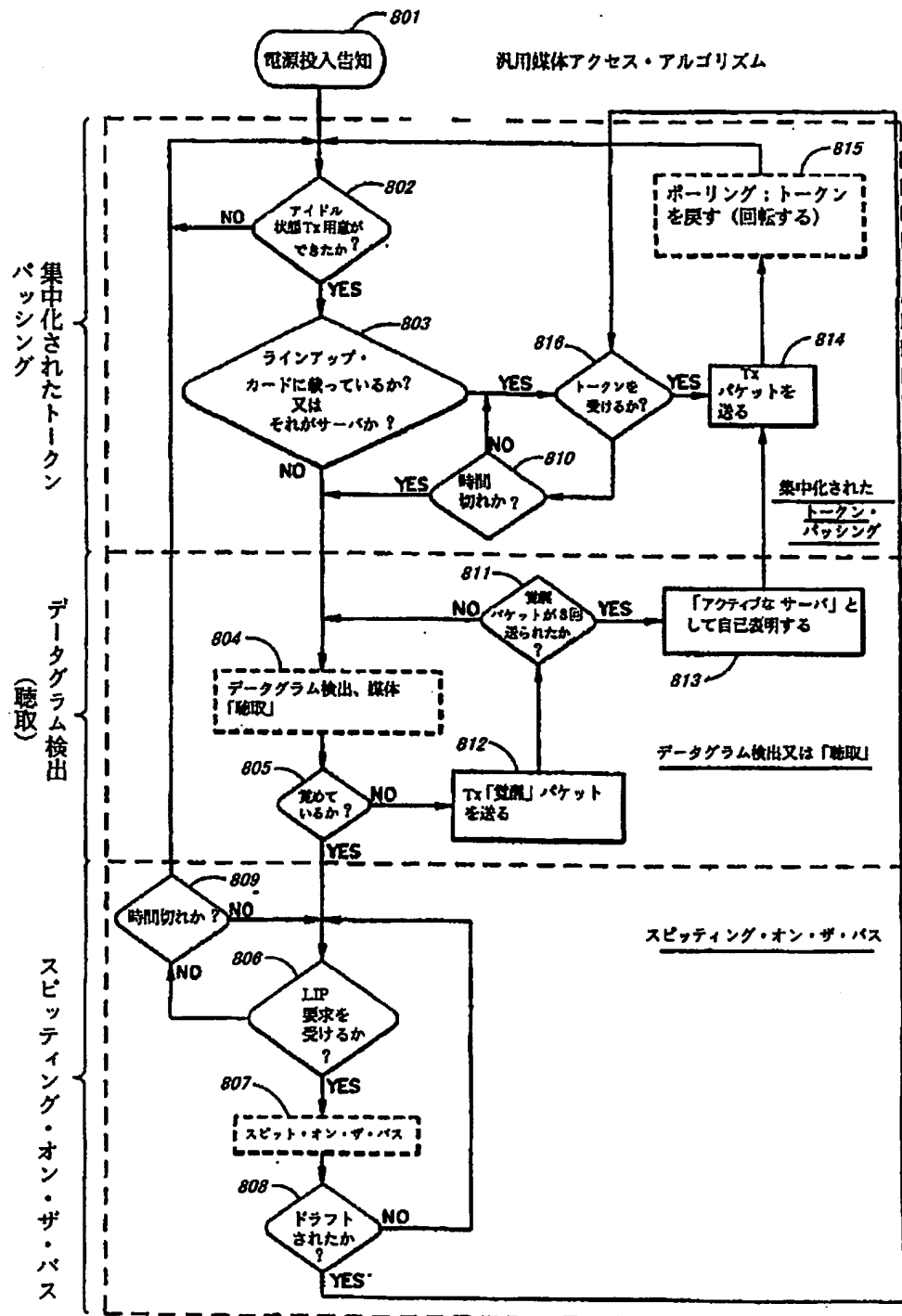
【図6】



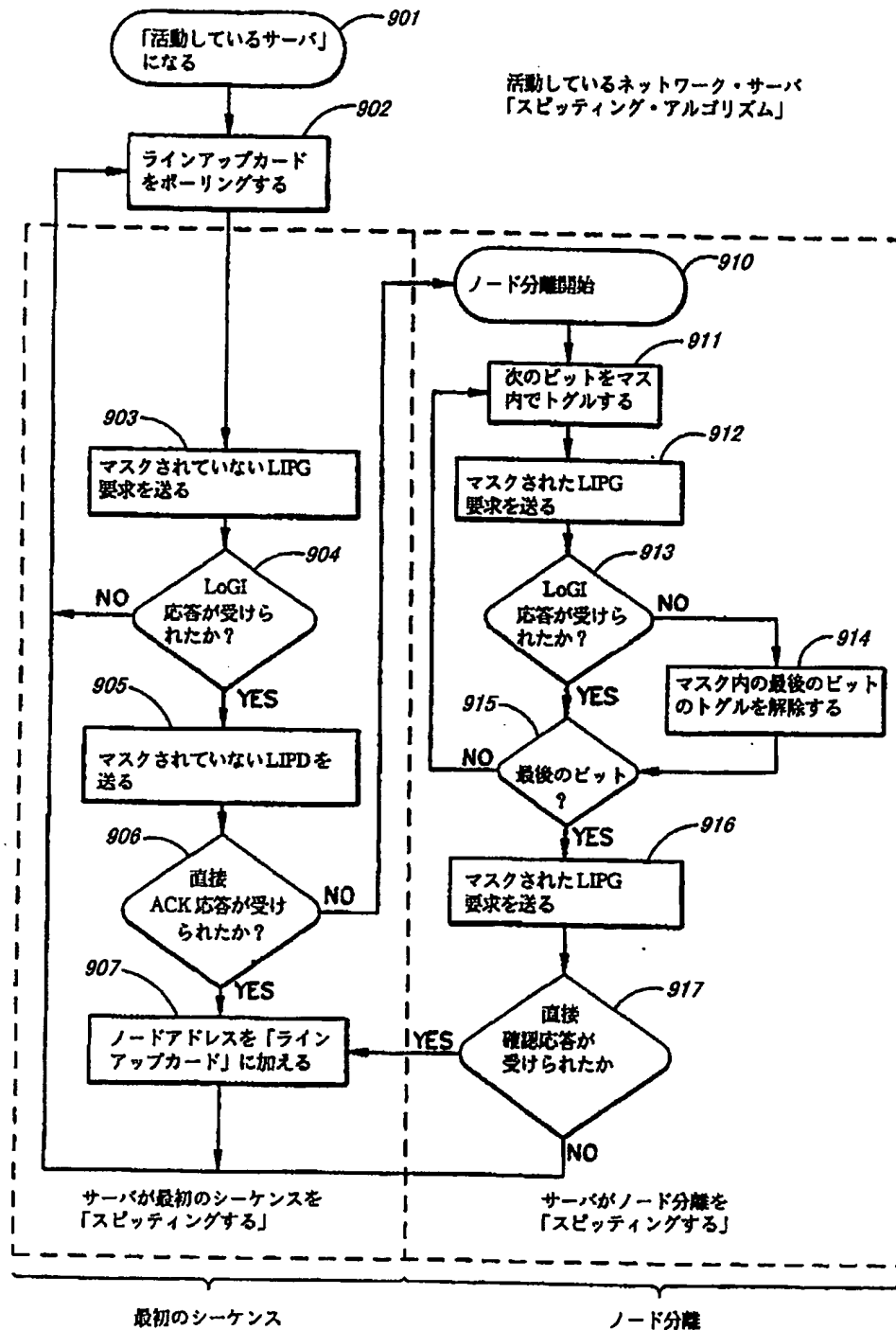
【図7】



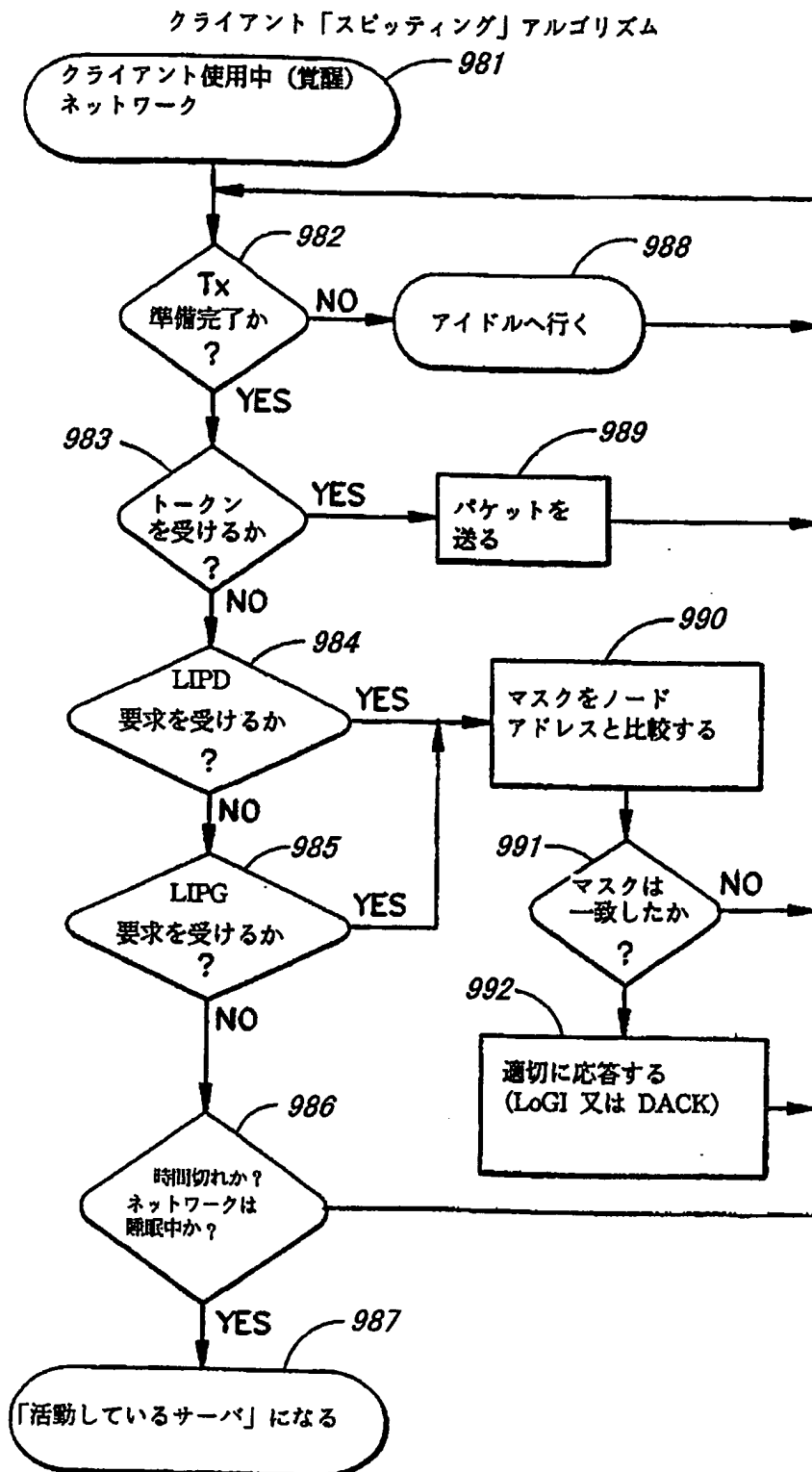
汎用媒体アクセス・アルゴリズム



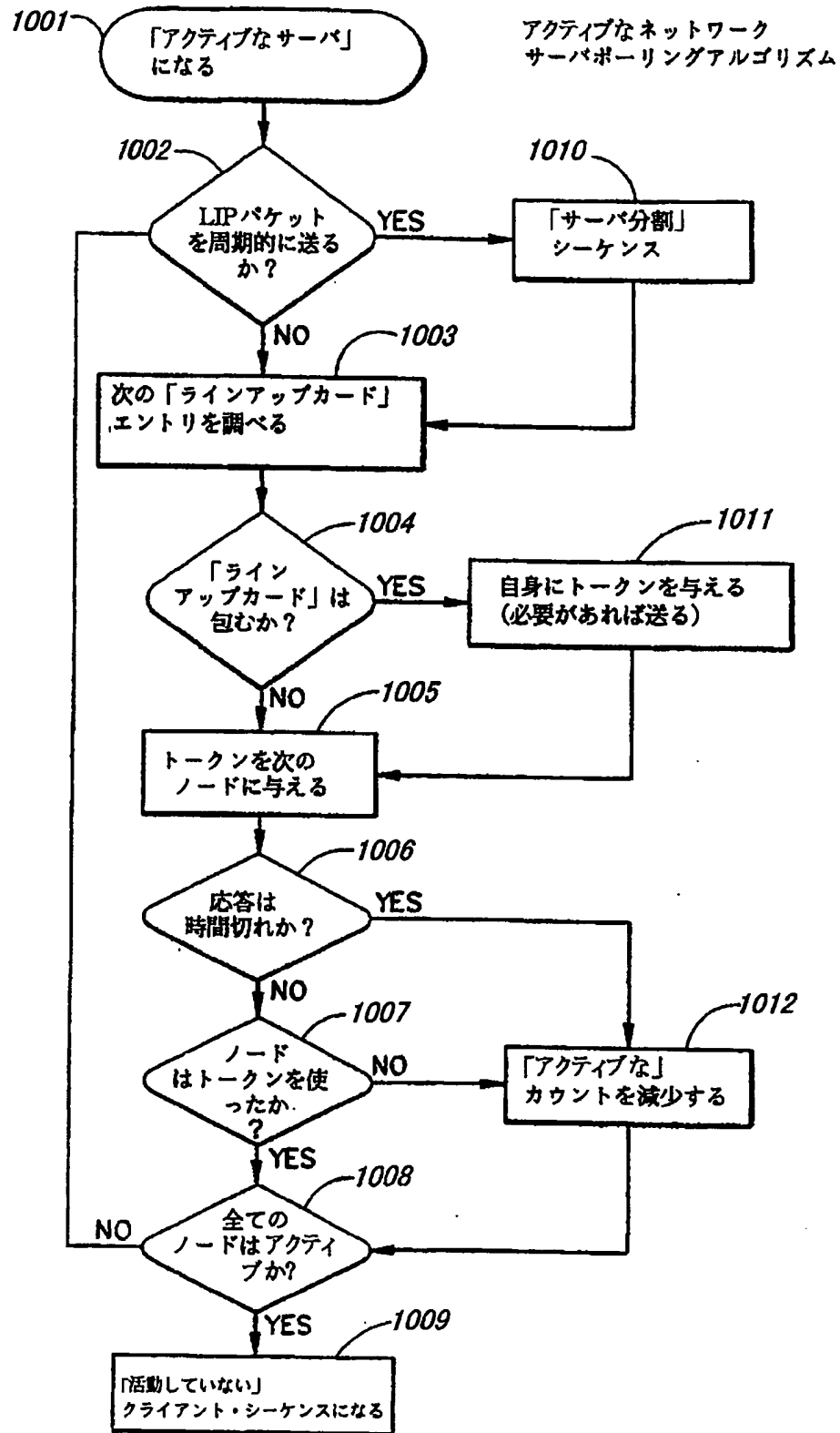
【図9A】



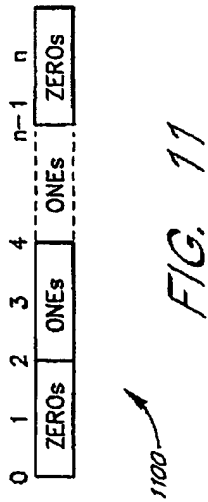
【図9B】



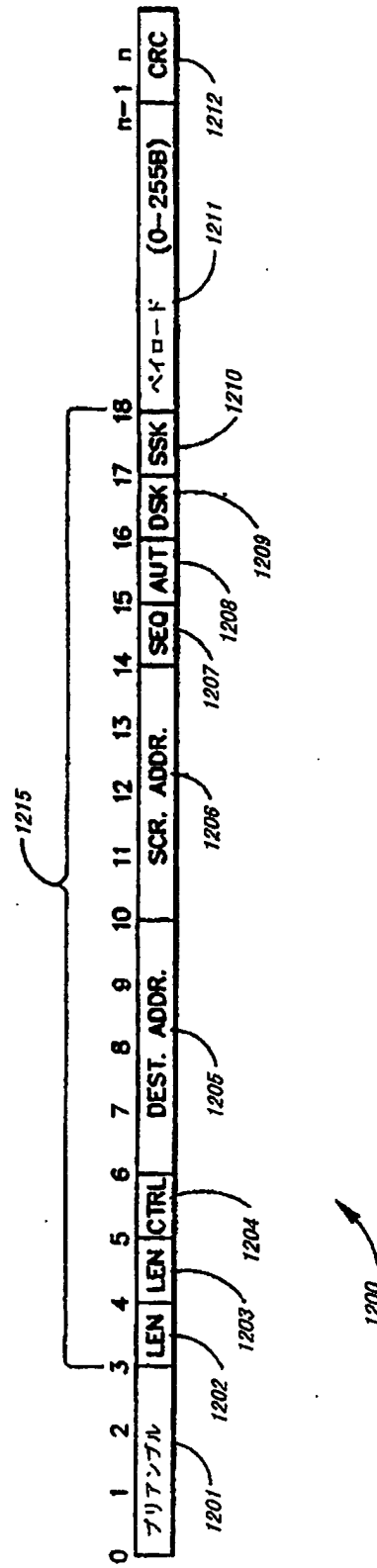
【図10】



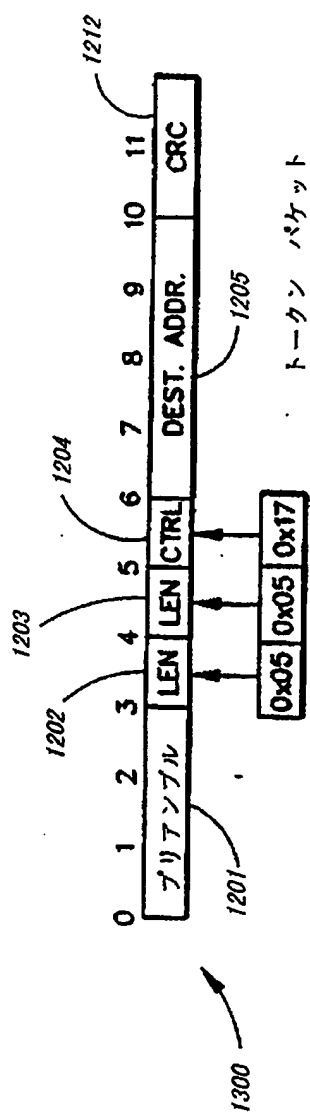
【図 11】



【図12】

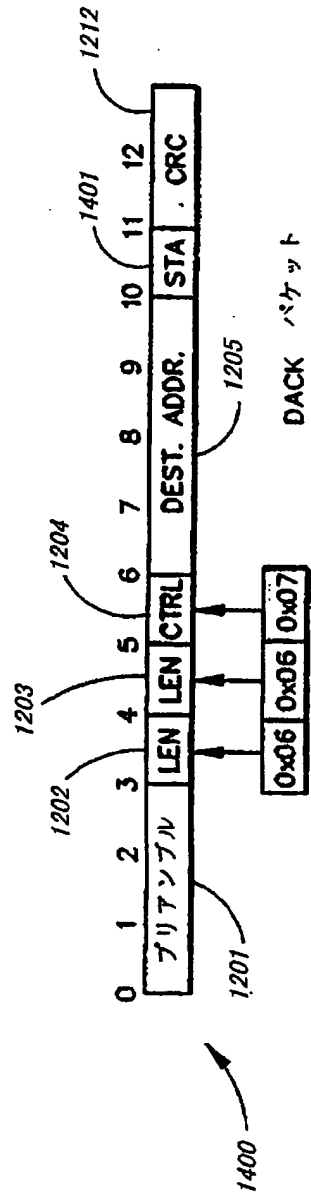


【図13】

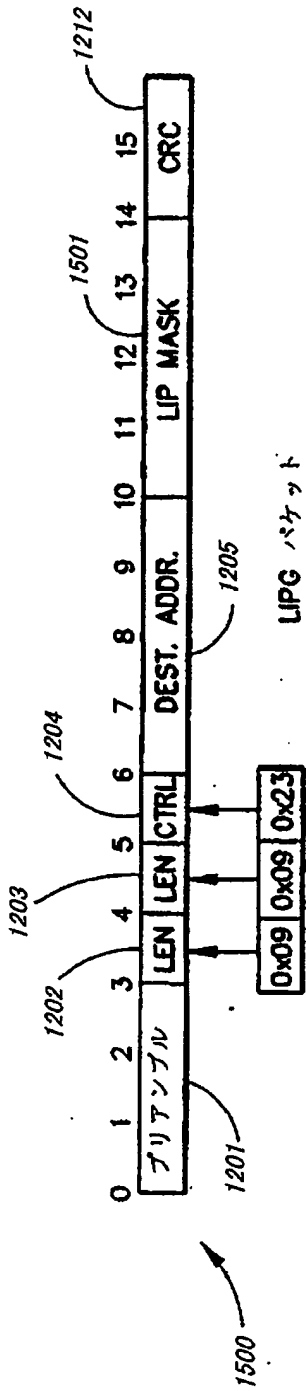




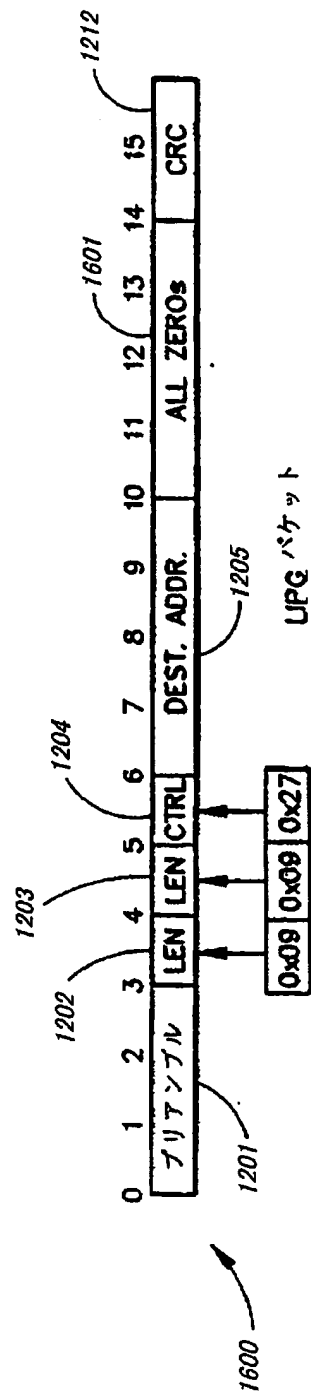
【図 14】



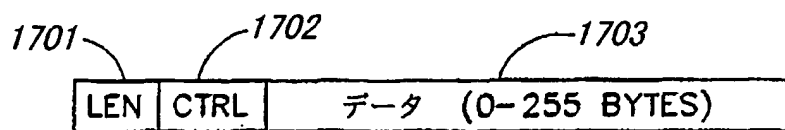
【図15】



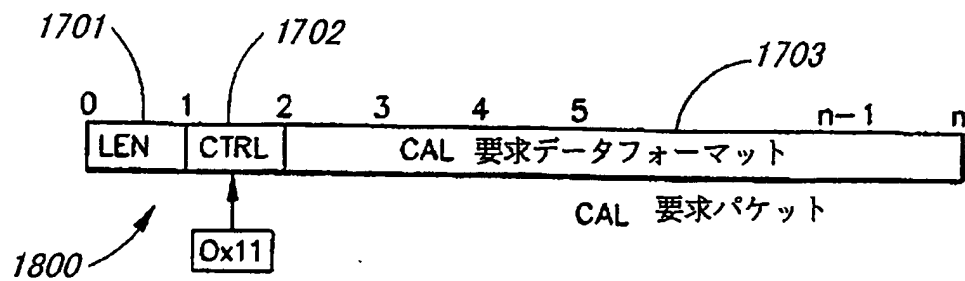
【図16】



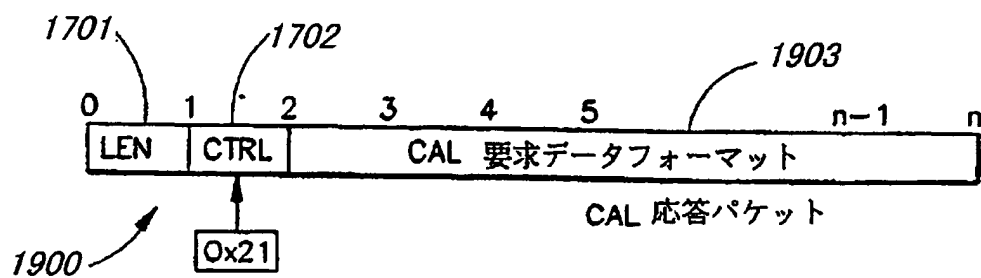
【図17】



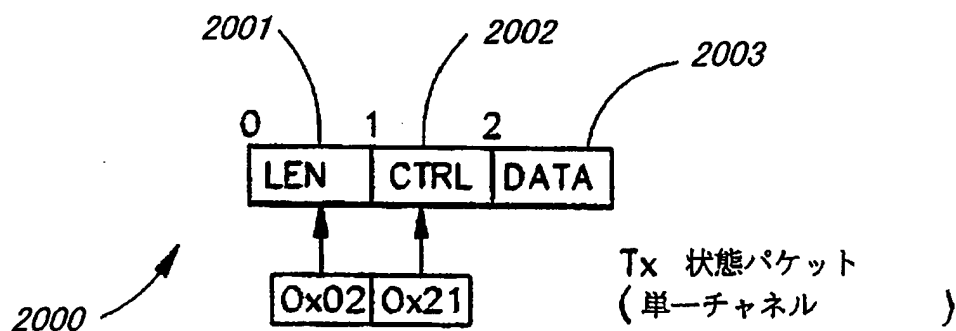
【図18】



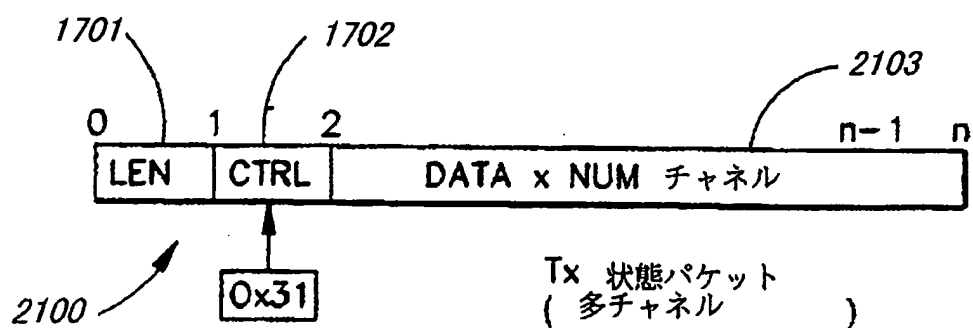
【図19】



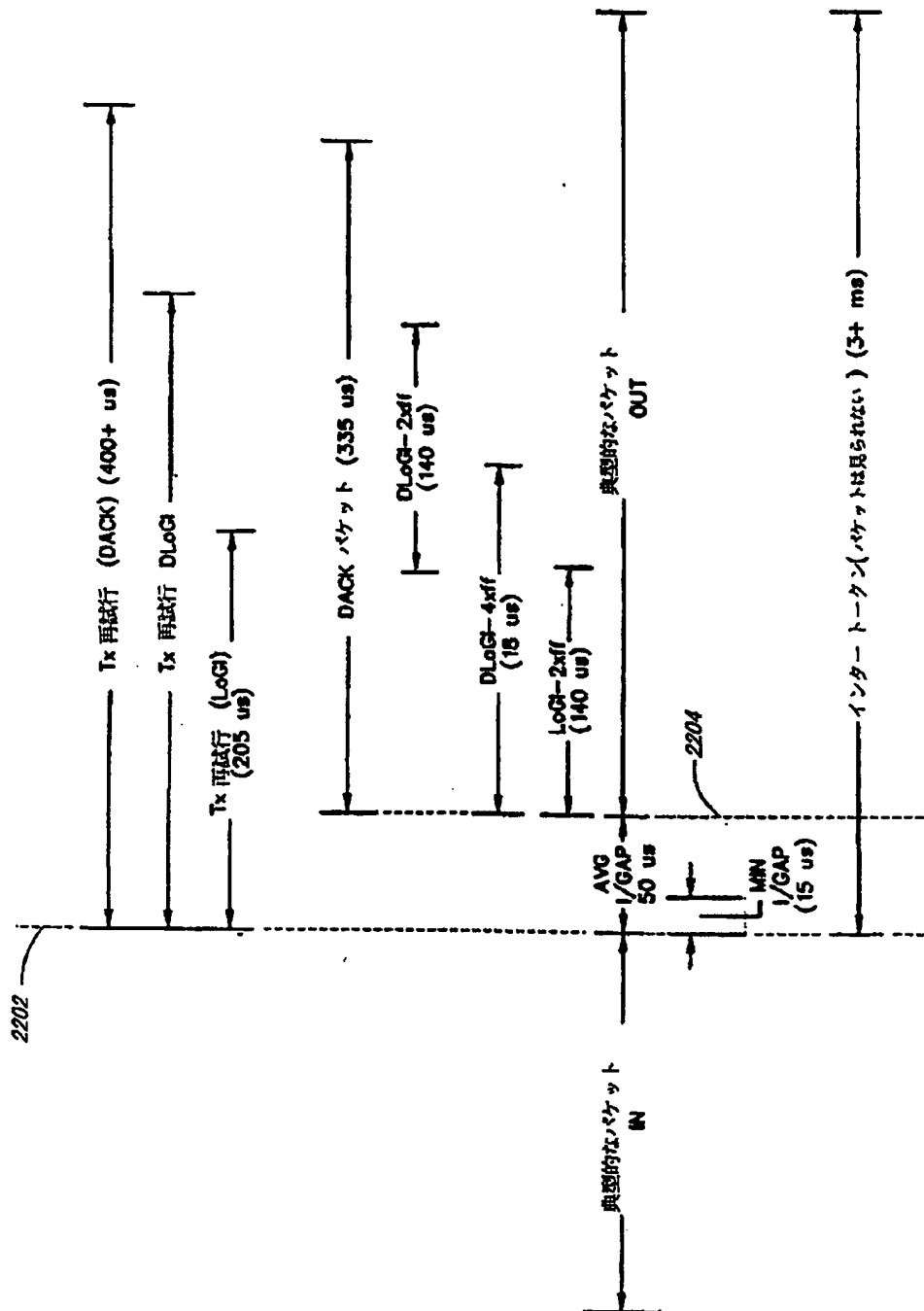
【図20】



【図21】



【図 22】



## 【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/US99/01234
<b>A. CLASSIFICATION OF SUBJECT MATTER</b> IPC(6) : G06F 13/14; G06F 13/42 US CL : 709/249 According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b> Minimum documentation searched (classification system followed by classification symbols) U.S. : 709/249, 201, 220, 224, 230; 710/1,20,31,33. Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched APS US PATENT SEARCH Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,608,720 A (BIEGEL et al.) 04 March 1997, Column 20, Lines 1-67, Column 36, Line 48 - Column 38, Line 10, Column 39, Line 35 - Column 42, Line 3.	1-5,7-13,19 ,21- 23,25-31
X	US 5,630,204 A (HYLTON et al.) 13 MAY 1997. Column 5, Lines 24-29, Column 15, Line 20 - Column 16, Line 43, Column 17, Line 3 - Column 18, Line 63, Column 40, Line 50 - Column 41, Line 61, Column 42, Line 44 - Column 43, Line 13.	17,18 —
Y	US 5,640,446 A (EVERETT et al.) 17 JUNE 1997, Column 6, Line 48 - Column 10, Line 46.	8, 9, 11, 12 ,26,27,29,30
Y	US 5,640,446 A (EVERETT et al.) 17 JUNE 1997, Column 6, Line 48 - Column 10, Line 46.	1-5,7-13,19 ,21- 23,25-31
Y	US 5,371,852 A (ATTANASIO et al.) 06 DECEMBER 1994, Column 7, Line 14 - Column 8, Line 42.	1,8,10,19,26,28
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents:	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is considered with one or more other such documents, such combination being obvious to a person skilled in the art *A* document member of the same patent family	
*A* document defining the general state of the art which is not considered to be of particular relevance *B* earlier document published on or after the international filing date *L* document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search 23 APRIL 1999	Date of mailing of the international search report 25 MAY 1999	
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer MARC THOMPSON <i>James R. Thompson</i> Telephone No. (703) 305-3900	

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US99/01234

## C (Continuation), DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,668,986 A (NILSEN et al.) 16 SEPTEMBER 1997, Column 2, Lines 34-36, Column 3, Lines 19-41.	1,19
Y,P	US 5,758,052 A (GLOWNY et al.) 26 MAY 1998, Column 3, Line 4 - Column 4, Line 13.	1,8,9,19,26,27
A,P	US 5,889,954 A (GESSEL et al.) 30 MARCH 1999, Entire document.	1-33
Y,P	US 5,796,999 A (AZAGURY et al.) 18 AUGUST 1998, Column 4, Line 29 - Column 6, Line 17.	1,19
Y,P	US 5,774,668 A (CHOQUIER et al.) 30 JUNE 1998, Column 8, Line 6 - Column 12, Line 4.	1,19
A,P	US 5,742,774 A (AL-SALAMEH et al.) 21 APRIL 1998, Entire document.	1-33
Y,P	US 5,893,118 A (SONDEREGGER) 06 APRIL 1999, Column 10, Lines 48-57.	1,19,14,15,32
A	US 4,853,843 A (ECKLUND) 01 AUGUST 1989, Entire document.	1-33

フロントページの続き

(81)指定国 EP(AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG), AP(GH, GM, KE, LS, MW, SD, SZ, UG, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW

Fターム(参考) 5B089 GA31 GB02 GB04 JA35 KA11

KA13 KF05 KF06

5K033 AA09 BA04 CB01 CB02 CB08

CB14 CC01 DA05 DB12 DB18

DB23 EC03

5K046 PS00 PS23 PS36

【要約の続き】

ほとんどいかなるレガシイ・データ・ネットワーク化サービスを電源線を通じて送ることができる。



【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成18年4月6日(2006.4.6)

【公表番号】特表2002-501251(P2002-501251A)

【公表日】平成14年1月15日(2002.1.15)

【出願番号】特願2000-528920(P2000-528920)

【国際特許分類】

G 0 6 F 13/00 (2006.01)

H 0 4 B 3/54 (2006.01)

H 0 4 L 12/28 (2006.01)

【F I】

G 0 6 F 13/00 3 5 7 A

G 0 6 F 13/00 3 5 1 B

H 0 4 B 3/54

H 0 4 L 12/28 1 0 0 H

H 0 4 L 12/28 2 0 0 M

【手続補正書】

【提出日】平成18年1月17日(2006.1.17)

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】請求項19

【補正方法】変更

【補正の内容】

【請求項19】 ネットワーク上のノードについての情報を含んでいるノード・データベースを作成することと、

前記ノード・データベースをアクセスする1つまたは複数のアクセス法を提供し、前記ノード・データベースを維持するゲートウェイを指定することと、

前記ノード・データベースを1つまたは複数の待機サーバ・ノード内に鏡像コピーすることと、

を有する、ネットワーク上のノード間で通信するために所望のプロトコルを使用する方法

。